

MDC E0857

CR-133965

(NASA-CR-133965) DEVELOPMENT OF
SIMULATION COMPUTER COMPLEX SPECIFICATION
Final Report (McDonnell-Douglas
Astronautics Co.) 195 p HC \$11.75

N73-27177

Unclas

CSCL 14B G3/11 09224

DEVELOPMENT OF SIMULATION COMPUTER COMPLEX SPECIFICATION

FINAL REPORT

MCDONNELL DOUGLAS ASTRONAUTICS COMPANY • EAST

Reproduced by
**NATIONAL TECHNICAL
INFORMATION SERVICE**
US Department of Commerce
Springfield, VA. 22151

MCDONNELL DOUGLAS



209 P. 0

COPY NO. 23

DEVELOPMENT OF SIMULATION COMPUTER COMPLEX SPECIFICATION

29 JUNE 1973

MDC E0857

FINAL REPORT

PREPARED FOR NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
LYNDON B. JOHNSON SPACE CENTER
HOUSTON, TEXAS 77058

SUBMITTED UNDER CONTRACT NAS 9-12882

MCDONNELL DOUGLAS ASTRONAUTICS COMPANY - EAST

Saint Louis, Missouri 63166 (314) 232-0232

MCDONNELL DOUGLAS



DEVELOPMENT OF SIMULATION COMPUTER COMPLEX SPECIFICATION

CONTENTS

	LIST OF FIGURES -----	v
	LIST OF TABLES -----	vii
	GLOSSARY -----	viii
Section 1	SUMMARY -----	1-1
Section 2	INTRODUCTION -----	2-1
Section 3	MISSION SIMULATOR CONFIGURATION -----	3-1
Section 4	SIMULATION SOFTWARE SIZING - TASK 1.1 -----	4-1
	4.1 Simulation Software Sizing Methodology -----	4-3
	4.2 Simulation Software Module Definition -----	4-7
	4.3 Data Sources -----	4-27
	4.4 Simulation Software Sizing Estimate - Conventional Approach -----	4-31
	4.5 Simulation Software Sizing Estimates - Advanced Techniques Approach -----	4-39
	4.6 Shuttle Flight Software - Sizing and Implementation ----	4-46
	4.7 Batch and Interactive User Processing Loads -----	4-56
	4.8 Summation of Sizing Data -----	4-58
Section 5	COMPUTER HARDWARE REQUIREMENTS - TASK 1.2 -----	5-1
	5.1 Simulation Software Implementation -----	5-2
	5.2 Simulation Software Load Analysis -----	5-9
	5.3 Requirements Specification - The Shuttle Mission Simulator Simulation Operations Mix -----	5-27
	5.4 Computer Simulation Analysis -----	5-35
	5.5 Statement of Computer Hardware Requirements -----	5-37
Section 6	DATA CONVERSION EQUIPMENT REQUIREMENTS - TASK 1.3 -----	6-1
	6.1 Host Computer to Simulator DCE -----	6-2
	6.2 Host Computer to Flight Computer DCE -----	6-14
	6.3 Simulator Flight Graphics -----	6-22
	6.4 Graphics Display Terminal For Instructor Station -----	6-25
	6.5 Host Computer to Mission Control Center Interface -----	6-28
	6.6 Operational Verification -----	6-29
	6.7 General Maintainability and Environmental Requirements -	6-30
	6.8 Acceptance Testing -----	6-31
	6.9 Reliability -----	6-32

CONTENTS (Continued)

Section 7	FLIGHT COMPUTER EMULATION REQUIREMENTS -----	7-1
	7.1 Characteristics of Microprogrammed Computers -----	7-1
	7.2 Requirements for a Shuttle Flight Computer Emulator ----	7-5
Section 8	SYSTEM SOFTWARE REQUIREMENTS - TASK 1.4 -----	8-1
	8.1 Generic System Software Requirements -----	8-2
	8.2 General Simulation Support Requirements -----	8-11
	8.3 Shuttle Mission Simulator Load Dependent Requirements --	8-18
	8.4 Requirement Modifications for Two Simulators -----	8-20
Section 9	SIMULATION MANAGEMENT PLAN - TASK 1.5 -----	9-1
	9.1 Task Descriptions -----	9-2
	9.2 Computer Complex Hardware/Software Requirements -----	9-9
Section 10	CONCLUSIONS AND RECOMMENDATIONS -----	10-1
	10.1 Conclusions -----	10-1
	10.2 Recommendations -----	10-3
Section 11	REFERENCES -----	11-1

LIST OF PAGES

Title Page

ii thru ix

1-1 thru 1-3

2-1

3-1 thru 3-4

4-1 thru 4-62

5-1 thru 5-53

6-1 thru 6-32

7-1 thru 7-8

8-1 thru 8-20

9-1 thru 9-11

10-1 thru 10-3

11-1 thru 11-2

DEVELOPMENT OF SIMULATION COMPUTER COMPLEX SPECIFICATION

FIGURES

<u>Number</u>		<u>Page</u>
3-1	Mission Simulator Configuration -----	3-2
4-1	Shuttle Mission Simulator Software Module Identification Chart -	4-8
4-2	Reference Computer MIPS vs Mission Phase - Conventional Approach - Configuration 1 -----	4-59
4-3	Reference Computer Core Size vs Mission Phase - Conventional Approach - Configuration 1 -----	4-59
4-4	Reference Computer MIPS vs Mission Phase - Advanced Technique Approach - Configuration 1 -----	4-60
4-5	Reference Computer Core Size vs Mission Phase - Advanced Technique Approach - Configuration 1 -----	4-60
4-6	Reference Computer MIPS vs Mission Phase - Conventional Approach - Configuration 3 -----	4-62
4-7	Reference Computer Core Size vs Mission Phase - Conventional Approach - Configuration 3 -----	4-62
5-1	I/O Requirements Summary -----	5-6
5-2	Functional Timeline of SMS Computation Process -----	5-8
5-3	Simulation Software Distribution Summary - Ascent Mission Phase - Single CPU Configuration -----	5-11
5-4	Simulation Software Distribution Summary - On-Orbit Mission Phase - Single CPU Configuration -----	5-12
5-5	Simulation Software Distribution Summary - Ascent Mission Phase - Dual CPU Configuration - CPU No. 1 -----	5-14
5-6	Simulation Software Distribution Summary - Ascent Mission Phase - Dual CPU Configuration - CPU No. 2 -----	5-15
5-7	Simulation Software Distribution Summary - On-Orbit Mission Phase - Dual CPU Configuration - CPU No. 1 ----	5-16
5-8	Simulation Software Distribution Summary - On-Orbit Mission Phase - Dual CPU Configuration - CPU No. 2 ----	5-17
5-9	Simulation Software Distribution Summary - Ascent Mission Phase - Four CPU Configuration - CPU No. 1 -----	5-19

FIGURES (Continued)

<u>Number</u>		<u>Page</u>
5-10	Simulation Software Distribution Summary - Ascent Mission Phase - Four CPU Configuration - CPU No. 2 -----	5-20
5-11	Simulation Software Distribution Summary - Ascent Mission Phase - Four CPU Configuration - CPU No. 3 -----	5-21
5-12	Simulation Software Distribution Summary - Ascent Mission Phase - Four CPU Configuration - CPU No. 4 -----	5-22
5-13	Simulation Software Distribution Summary - On-Orbit Mission Phase - Four CPU Configuration - CPU No. 1 ---	5-23
5-14	Simulation Software Distribution Summary - On-Orbit Mission Phase - Four CPU Configuration - CPU No. 2 ---	5-24
5-15	Simulation Software Distribution Summary - On-Orbit Mission Phase - Four CPU Configuration - CPU No. 3 ---	5-25
5-16	Simulation Software Distribution Summary - On-Orbit Mission Phase - Four CPU Configuration - CPU No. 4 ---	5-26
6-1	Host Computer to Simulator DCE -----	6-4
6-2	Minicomputer to Analog/Digital Interface -----	6-7
6-3	Host Computer to Flight Computer DCE -----	6-16

TABLES

<u>Number</u>		<u>Page</u>
4-1	Simulation Software Modules -----	4-9
4-2	Simulation Software Modules Affected by Redundancy -----	4-26
4-3	Summary of Modules vs Source Simulation -----	4-27
4-4	Software Module Sizing Estimates - Conventional Approach -----	4-33
4-5	Aerodynamic Data Storage Requirements -----	4-44
4-6	Conventional Techniques/Advanced Techniques Sizing Estimates -	4-45
4-7	Summary of Flight Software Sizing and Flight Computer Loading	4-48
4-8	ICS Sizing and Timing Data for Existing Simulators -----	4-52
5-1	Shuttle Mission Simulator Software Summary -----	5-3
5-2	Software Used to Define FORTRAN Operations and Simulation Operations Mis -----	5-28
5-3	Shuttle Mission Simulator Simulation Software Operations Mix -	5-30
5-4	Comparison of Gibson III Mix and Shuttle Mission Simulator Operations Mix -----	5-31
5-5	Central Processing Unit Simulation Software Loading -----	5-39
5-6	Computer Hardware Minimum Configuration Requirements -----	5-52
7-1	Requirements for Emulator Computer -----	7-8
8-1	High Frequency Executions of System Software Functions -----	8-19
8-2	Low Frequency Executions of System Software Functions -----	8-19
9-1	Computer Complex Hardware/Software Requirements Summary -----	9-10

GLOSSARY

A/D	Analog to Digital
ADI	Analog to Digital Inputs
ANSI	American National Standards Institute
BPS	Bits Per Second
CDC	Control Data Corporation
CIG	Computer Image Generation
CMPS	Command Module Procedures Simulator
CMS	Command Module Simulator
COBOL	Common Business Oriented Language
COMPASS	CDC Computer Assembly Language for 6000/7600
COMPSIM	Computer Simulation Program
COUNTR	A Computer Program for Analyzing Assembly Language Programs
CPM	Cards per Minute
CPU	Central Processing Unit
CRT	Cathode Ray Tube
D/A	Digital to Analog
DCE	Data Conversion Equipment
DMA	Direct Memory Access
DP&S	Data Processing and Software Subsystem
D/R	Digital to Resolver
DSI	Discrete Input
DTC	Data Transfer Command
FDAI	Flight Director Attitude Indicator
FS	Full Scale
GN&C	Guidance Navigation and Control
HOL	Higher Order Language
Host Computer	The Central Computer in the Training Simulation Complex
Hz	Hertz (i.e., cycles per second)
IBM	International Business Machines
ICS	Interpretive Computer Simulation
I/O	Input/Output

GLOSSARY (Continued)

LOADEM	A Computer Program Which Developes the Software Load Distribution
LPM	Lines Per Minute
LSB	Least Significant Bit
LSI	Large Scale Integration
ma	Milliamps
MCAUTO	McDonnell Douglas Automation Company
MCC	Mission Control Center
MDAC	McDonnell Douglas Astronautics Company
MDC	McDonnell Douglas Corporation
MDEC	McDonnell Douglas Electronics Company
MIPS	Million Instructions Per Second
MOPS	Million Operations Per Second
ms	Millisecond
MTTF	Mean Time to Failure
OS	Operating System
PCM	Pulse Code Modulated
pf	Picofarads
PLH	Payload Handling Computer
PM	Performance Measurement Computer
PPU	Peripheral Processing Unit
PSYZR	A Computer Program Which Converts Source Sizing Data to Reference Computer Estimates and Sorts the Resulting Data
rad/sec	Radians Per Second
ROM	Read-Only Memory
SMS	Shuttle Mission Simulator
STS	Space Transportation System
TSCC	Training Simulation Computer Complex
VDC	Volts Direct Current
WPS	Words Per Second
η seconds	Nanoseconds
μ seconds	Microseconds

Section 1

SUMMARY

This Final Report summarizes the results of all of the Tasks performed in the course of defining requirements for the Training Simulation Computer Complex. These Tasks consisted of the Simulation Software Sizing, Task 1.1; Computer Requirements, Task 1.2; Data Conversion Equipment Requirements, Task 1.3; System Software Requirements, Task 1.4; and the Simulation Management Plan, Task 1.5. The scope of this report is summarized below.

The principal results of the Simulation Software Sizing Task were the derived estimates of the computer loadings attributable to the Shuttle Mission Simulator Simulation Software, the Shuttle Flight software, and the associated batch and interactive user processing load. These data represent the total software load, except for computer system software, that would be imposed on the computer complex by the Shuttle Mission Simulator. Implementation of the flight software with either interpretive simulation of the flight computers or functional simulation of the software was also evaluated. Use of flight computers or flight computer emulators seemed desirable as a means of limiting the host computer processing loads to a reasonable level. A computer program, PSYZR, was developed for summing, sorting, and listing the software modules. This program greatly facilitates the generation of alternate software loadings on a consistent and comprehensive basis that reflect desired variations in module characteristics, computation frequency, active mission phase, etc. The Simulation Software Sizing, Task 1.1, is summarized in Section 4.

The Simulation Software Sizing results were analyzed further during the Computer Requirements Task 1.2. Factors considered were the I/O loadings associated with the processing load, the timewise sequencing of I/O and processing, the distribution of software between multiple processors when required, and the specification of processor performance in terms assuring equal processing capability from one computer to another. This last consideration prompted development of a simulation software FORTRAN operations mix to enable specification of processor performance in terms of millions of operations executed per second, MOPS.

The CPU requirements were then summarized as follows. The CPU must be capable of supporting the computing load for the worst case loading, the On-Orbit mission

DEVELOPMENT OF SIMULATION COMPUTER COMPLEX SPECIFICATION

phase. Up to ten percent of the available frame time will be allocated for completion of simulation input/output data transfers. Batch processing equivalent to .332 MIPS or .083 MOPS must be processed during the I/O period. For multiple CPU configurations, all CPU's within a configuration must be of the same capacity to allow flexibility of design and software reconfiguration potential. The CPU processing requirements are summarized below:

<u>CPU Requirements</u>		
<u>Configuration</u>	<u>MIPS</u>	<u>MOPS</u>
Single CPU	8.4	2.1
Dual CPU (Each)	5.0	1.25
Four CPU (Each)	3.1	0.78

System Software processing loads are to be added to these simulation requirements.

The Computer Requirements Task, summarized in Section 5, also established memory requirements, secondary storage requirements, and basic computer complex configuration requirements as well as requirements for microprogrammed computers to emulate the flight computers, summarized in Section 7.

The Data Conversion Equipment Requirements Task reviewed the anticipated simulator configuration and established requirements for each of the major data paths. Requirements for data conversion equipment were defined for the data paths between the host computer and the following simulator hardware: Mission Control Center, flight computers, instructor displays, and flight hardware graphics. Conventional minicomputers are specified for data formatting and data distribution to the simulator hardware, simulator graphics displays, and instructor display terminals. Special purpose, microprogrammed minicomputers were specified as interfaces to the five flight computers. Also defined were the requirements for the data conversion components which interface the minicomputers to the simulator hardware and the host computer. The DCE requirements are summarized in Section 6.

The System Software Requirements Task defined the functional features for the system software and also the numbers of system functions which must be processed concurrent with the simulation load.

FINAL REPORT

DEVELOPMENT OF SIMULATION COMPUTER COMPLEX SPECIFICATION

MDC E0857
29 JUNE 1973

It was determined that the most comprehensive currently available system software is required, with certain modifications and additions necessary for operation in a real time simulation environment. The operating system will require real-time monitors which minimize the overhead associated with the high task switching rates necessitated by the 25 frame per second operating requirements. The operating system will require special routines for simulator hardware communication and initialization, for bit-level manipulation, for simulator and interface checkout, and for instructor display control and communication.

The operating system must provide for full multi-programmed operation to perform simultaneous real-time simulation, batch and time-sharing operation. It must provide a comprehensive job processing system, accounting system and operator control capability. It must provide maximum user assistance in file assignment and manipulation, error processing and I/O control. These capabilities are generally available as has been established by the Background Survey Task. System Software Requirements are summarized in Section 8.

The Simulation Management Plan Task, Section 9, defined a simulation management plan to expedite and monitor the procurement, development, implementation and acceptance of the Shuttle Mission Simulator Complex at the Lyndon B. Johnson Space Center. The primary objective of this Task was to determine if any additional hardware and system software requirements existed for the computer complex. A limited number of hardware and software requirements were identified although the requirements for many of these features aren't limited to implementation of a management plan. The plan itself presents an approach to simulator implementation that might be followed by a contractor, such as MDAC.

The results of all of the above Tasks are summarized in this report and are presented in considerably greater detail in reports previously published. The requirements presented herein have been incorporated in specifications for the Training Simulation Computer Complex. The Background Survey Task has identified candidate vendor systems that may satisfy these requirements and evaluated them accordingly.

It is recommended that the results of this study be used in support of the procurement of a Training Simulation Computer Complex.

DEVELOPMENT OF SIMULATION COMPUTER COMPLEX SPECIFICATION

Section 2
INTRODUCTION

This report presents the Computer Complex Requirements defined during the Training Simulation Computer Complex Study being conducted for the Johnson Space Center of the National Aeronautics and Space Administration under Contract Number NAS 9-12882. This study was performed by McDonnell Douglas Astronautics Company - East at its Houston Operations facility with subcontract support from the TRW Systems Group. David D. Lang was Technical Manager for the contract for the NASA. Theodore H. Wenglinski was Principal Investigator for the study for McDonnell Douglas Astronautics Company. E. Donald Stuckle was technical coordinator for TRW for their subcontracted support for this study.

The Training Simulation Computer Complex Study was one of three studies contracted in support of Johnson Space Center preparations for procurement of a Shuttle Mission Simulator for Shuttle crew training. The subject study was concerned with definition of the software loads to be imposed on the computer complex to be associated with the Shuttle Mission Simulator and the development of procurement specifications based on the resulting computer requirements. These procurement specifications cover the computer hardware and system software as well as the data conversion equipment required to interface the computer to the simulator hardware.

The development of the necessary hardware and software specifications required the execution of a number of related tasks which included Simulation Software Sizing, Computer Requirements definition, Data Conversion Equipment Requirements definition, System Software Requirements definition, a Simulation Management Plan, a Background Survey, and finally preparation of the specifications. Results of the Simulation Software Sizing Task, The Computer Complex Hardware Requirements, the System Software Requirements and The Simulation Management Plan have been published, The Background Survey, and the specifications are being published concurrently with this report.

This Final Report summarizes the results of the requirements Tasks and consequently presents all of the requirements in one consolidated document for the first time. These requirements are the basis for the specifications being published concurrently.

Section 3

MISSION SIMULATOR CONFIGURATION

A nominal mission simulator configuration has been required throughout the Training Simulator Computer Complex Study to assure that all analyses considered all aspects of the simulator requirements on a consistent basis. A comprehensive simulator configuration was therefore defined without any extensive configuration analyses. Instead, the modularity of the data developed for this simulator configuration and the visibility provided by the study documentation should facilitate adjustment of the computer requirements for whatever changes in simulator or Shuttle configuration are later defined.

The Shuttle Mission Simulator configuration, shown in Figure 3-1, reflects the current Shuttle avionics configuration. The flight computers consist of three GN&C and two auxiliary computers for payload management and payload manipulation. Flight computers associated with the main engine control loops are not shown since a functional simulation of the performance of those units should be adequate for generating any signals perceivable to the crew from normal operation or malfunction.

The configuration shown also assumes full implementation of all crew positions on the motion base. The modularity, detail, and completeness of the data presented are intended to facilitate assessment of multiple crew station configurations as required. The configuration, as shown, provides for direct control of the crew station hardware, including CRT displays, as well as the flight computers from the host computer. There are no flight computer to flight computer or flight computer to instruments transfers of data as in the Shuttle vehicle. Control from the host computer provides more flexibility for introducing failures in the system. As an example, the failure of an instrument may be simulated by modifying the appropriate drive signals from the host computer. Modifying the drive signals to the instruments would be very difficult, if not impossible in some instances, if the instruments were driven by the flight computers. For the configuration shown, environment and dynamic data computed in the host computer, are transmitted to the flight computers. All flight computer outputs are returned to the host computer, where they may be modified for malfunction insertion, and then transmitted to the flight graphics, instructor station, crew stations, and other flight computers.

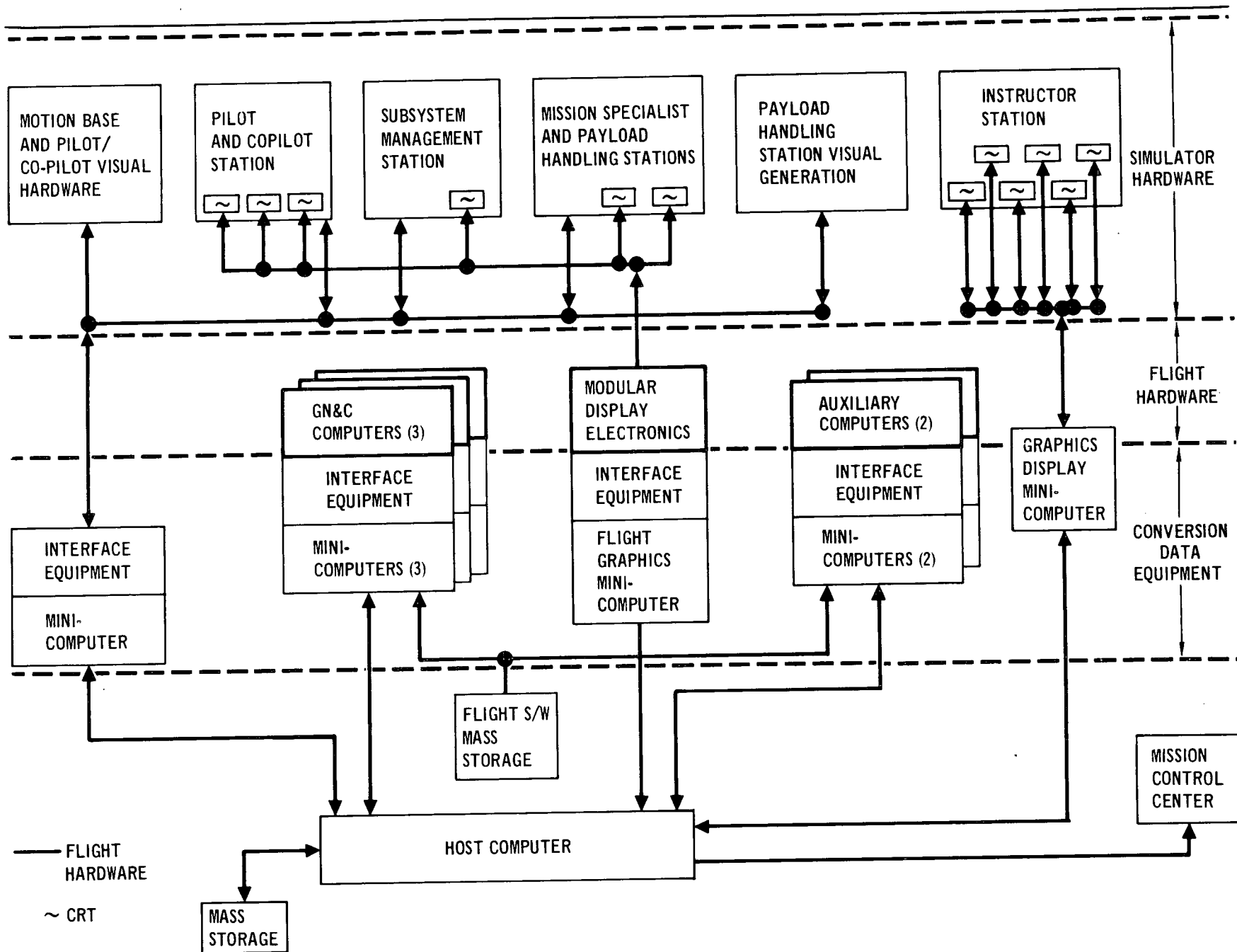


Figure 3-1 Mission Simulator Configuration

FINAL REPORT

DEVELOPMENT OF SIMULATION COMPUTER COMPLEX SPECIFICATION

MDC E0857
29 JUNE 1973

The minicomputers shown act as buffers between the host computer and the flight computers, and the host computer and other simulator hardware. These buffers enable the host computer to transfer its data in continuous blocks at the beginning or end of a time frame and thus minimize interrupt processing by the host computers. It is anticipated that the flight computers will transfer data to and from the minicomputers throughout the time frame. The minicomputers will also format the data during the exchange. Minicomputers with very fast, read-only, control storage memories for specialized input/output control are required for the flight computer interfaces.

The basic unit of communication with the system from the instructor station is CRT Graphics displays and keyboards which provide maximum flexibility both for monitoring and for introducing inputs. Mode select switches are also provided for computer inputs. It is assumed that any instrumentation and displays that duplicate crew station hardware are to be wired in parallel directly from the crew stations and will not require additional computer interface hardware.

The out-the-window visual systems for the crew stations are assumed to be model and television systems with the exception of the payload handling visual scenes. The payload handling visual scene will be simulated either by Computer Image Generation (CIG) or by a model and television system. This is in accordance with recent findings in the visual study performed by MDEC for the NASA. It is assumed that the visual systems will require only digital information from the host computer. All encoding will be performed in the image generation equipment if used. Hand-controller signals and controls will be sent to the host computer. Image position and angle commands will be sent to the visual displays from the host computer. The image generation software is assumed to be contained in a special purpose computer that is part of the image generation equipment.

The data transmitted to the Mission Control Center will be in the form of Pulse Code Modulated (PCM) serial data. This requires data formatting that may be accomplished either in the host computer or a secondary computer.

The configuration shown in Figure 3-1 and the basis for the data presented herein

FINAL REPORT

DEVELOPMENT OF SIMULATION COMPUTER COMPLEX SPECIFICATION

**MDC E0857
29 JUNE 1973**

represents the requirements for implementation of one complete Shuttle Mission Simulator with all crew stations fully simulated, with full visual displays and with a motion base. Simulator hardware limitations, such as motion base weight restrictions, may preclude assembly of a comprehensive configuration. However, the detailed data shown herein facilitate adjustment of the requirements for less complex configurations or combinations thereof.

Section 4

SIMULATION SOFTWARE SIZING - TASK 1.1

This section presents the results of the Task performed to estimate the software load which would be placed on the Training Simulation Computer Complex (TSCC) by one Shuttle Mission Simulator (SMS). The results of the Simulation Software Sizing Task have been documented fully in Reference 1. Supporting data for each module sized is available in Reference 2.

This task required the definition of the software modules required, the acquisition and analysis of similar modules from previous simulators, and the estimation of the corresponding software characteristics for the Shuttle Mission Simulator.

Simulation of the Shuttle, however, requires simulation of the Shuttle flight computers or their functions. Consequently, estimates were also derived for the characteristics of the Shuttle flight software. These results were then used to assess the possibilities for interpretive simulation of the flight computers and for functional simulation of the flight software.

The magnitude of the software load associated with the SMS made it of considerable interest to consider the impact on this load of the use of more complex or sophisticated simulation techniques. These concepts included sampled data techniques for reducing computational frequencies and sophisticated curve fit techniques for reducing data storage memory requirements. Alternate estimates of the total software load, based on use of these techniques, were projected.

The methodology developed and applied during this task is reviewed briefly in Section 4.1. The definitions of the simulation software modules that were sized, are presented in Section 4.2. The data sources used for sizing these modules are noted in Section 4.3. The results of sizing the simulation software assuming conventional simulation techniques are presented in Section 4.4. Section 4.5 summarizes the effects on the software loading of use of the more complex simulation techniques.

Flight software size estimates are presented in Section 4.6 along with the resulting

FINAL REPORT

DEVELOPMENT OF SIMULATION COMPUTER COMPLEX SPECIFICATION

MDC E0857
29 JUNE 1973

interpretive and functional simulation requirements.

Estimates of the anticipated batch load for the TSCC are reviewed in Section 4.7.
A summary of the simulation software sizing results is presented in Section 4.8.

The results discussed in these sections are described in substantially greater detail in Reference 1.

DEVELOPMENT OF SIMULATION COMPUTER COMPLEX SPECIFICATION

4.1 SIMULATION SOFTWARE SIZING METHODOLOGY

Estimation of the software loading to be imposed on the Training Simulation Computer Complex (TSCC) by the Shuttle Mission Simulator (SMS) required a number of different activities. The first of these was a review of the current Shuttle configuration and the definition of the simulation software modules required for the implementation of the SMS. The functional descriptions of these modules had to be established to assure provision of all required functions and minimize duplication of functions. The modules identified and their functional descriptions are presented in Section 4.2.

Following this definition of the required modules, it became possible to review software from existing simulators and identify data which was basically similar to SMS requirements. This source data is noted in Section 4.3. The source data, when available, was then used to make an estimate for a similar module for the SMS. Since the source data was implemented on any of a number of different computers, a standard reference computer had to be established and all data adjusted to this reference. The following paragraphs summarize, briefly, the estimation techniques, the definition of the reference computer, and the conversion of data to the reference computer.

The methods used for sizing the simulation software with the use of advanced simulation techniques and the analyses of the flight software and its simulation or implementation are noted in the sections where those results are presented.

4.1.1 Estimating Techniques

Several different techniques were defined and used through the Task in determining the sizing estimates for the various software modules. The techniques varied as a function of the availability and type of existing simulation data. Data was gathered from many sources and used as a basis for estimating the Shuttle Mission Simulator software requirements.

For those data sources where only FORTRAN listings and memory maps were available, memory requirements were determined from the memory maps. The maximum number of instructions executed per pass was estimated by studying the FORTRAN listings, looking for Do-loops and branches, and estimating the increase or decrease from the total storage requirements.

When the source data was in the form of assembly language code, instruction counts were made directly but again the program flow had to be studied for loops and branches and maximum instructions executed had to be estimated. When FORTRAN routines had the generated assembly code also printed, the same procedure was followed as for the pure assembly language programming data.

When actual FORTRAN source statements were available, in either card, tape or disc form, test compilations were made on the CDC 6400 computer in Building 35. The code resulting from the compilation was analyzed by a utility program, COUNTR, which was written to support this task. COUNTR analyzes CDC assembly language code and counts the number of each of 48 instruction types which it identifies. A complete description of the program is contained in Reference 2. This program provided an exact count of instructions required for source routines. To obtain the count of maximum instructions executed, an estimate was still required since COUNTR did not detect branches or analyze loops.

In some instances timing data was available for the source data routines. Using the timing data and a conversion factor for average instruction time, the maximum instructions executed per pass were directly available.

There were several modules for which there was no source simulation data available except for a mathflow of the required operations. In other cases, mathflows had to be developed from functional descriptions of the computational requirements. For these modules, an instruction estimating algorithm was used to determine the SMS sizing estimate.

Regardless of how the raw sizing data was obtained, the next step in the estimating process required a comparison of the functional requirements of the module of interest with those of the source simulation model. If required, the source simulation data was adjusted to account for additional features in the Shuttle module or to account for the fidelity requirements of the mission simulator.

In order to process all of the module data assembled a second utility program was written during the study. The program, PSYZR, is able to tabulate and sum the data, and do a number of useful sorts. A complete description of the program, PSYZR, is contained in Reference 2.

DEVELOPMENT OF SIMULATION COMPUTER COMPLEX SPECIFICATION

4.1.2 Definition of Reference Computer

A Reference Computer and Compiler were defined to enable conversion of all the software sizing data to one consistent basis. The Reference Computer selected is a CDC 6000 series type machine except that it does not pack more than one instruction per word. The Reference Computer speed characteristics are not specified (nor required) since the data reflecting speed requirements are presented as the number of instructions that have to be executed per second. The compiler defined for the Reference Computer is a FORTRAN IV compiler. All the software sizing estimates are presented for this computer/compiler combination.

The CDC 6000 series computer was selected as a reference primarily because of our familiarity with this type machine and the availability of a NASA-owned CDC 6400 computer for benchmark runs.

4.1.3 Conversion of Source Data to the Reference Computer

The parameters selected for collection of the simulation software sizing data have minimized the amount of conversion required to reach a common baseline as well as the amount of benchmark data required for definition of these conversion factors. The collection of data in terms of the number of instructions to be stored, the number of instructions executed per pass, the number of parameters communicated and the amount of data stored minimized the aspects of computer system performance that must be assessed.

In addition, the Reference Computer, defined in Paragraph 4.1.2, was selected to provide the most convenient common basis for correlation of sizing data from the various source simulations discussed in Paragraph 4.2. Consequently, reference data for 49 modules required no conversion other than unpacking of the instructions from memory word counts for those instances in which assembly listings of the source software were not available. The second largest amount of data was derived from simulations implemented on the DDP 124 computers. For this data a benchmark program was prepared and compiled in order to obtain conversion factors which would assure a high degree of confidence in the results. In addition, data has been derived from a number of simulators which use still other computers. Conversion factors for this data were derived from benchmark runs obtained from McDonnell Douglas Automation Company.

FINAL REPORT

DEVELOPMENT OF SIMULATION COMPUTER COMPLEX SPECIFICATION

MDC E0857
29 JUNE 1973

The conversion factors that required definition, consistent with the data correlation approach noted, consisted of the following:

- a) The ratio of machine language instructions required on the Reference Computer to the number obtained on the source computer.
- b) The average number of machine language instructions stored per word of memory for the source computers..

No factor was required with respect to the memory words required to store floating point data. This is attributable to the practice of counting or estimating numbers of parameters rather than using memory area requirements.

DEVELOPMENT OF SIMULATION COMPUTER COMPLEX SPECIFICATION

4.2 SIMULATION SOFTWARE MODULE DEFINITION

The major software modules required for the SMS are identified and presented in Figure 4-1. Expansion to a more detailed level for most modules was performed as apparent in Table 4-1. The purpose of Figure 4-1 was to provide a tool for insuring that all the software modules were identified. It does not, nor was it intended to, define the structural arrangement of software for the Shuttle Mission Simulator (SMS).

Functional descriptions of the modules are contained in Table 4-1. A more detailed discussion of each module is contained in Reference 2 which presents the sizing estimates derived for each module.

The Systems Software modules (inside the dashed lines of Figure 4-1) were sized only for those cases for which they were required to run in a real-time mode to support the Simulation Software (e.g., a rate gyro module may call one of the Subroutine Library subroutines such as the sine routine). Even in these cases, they were sized only for their contribution to the number of instructions that had to be processed each second and these numbers were added to the processing load of the module calling them. No allowance was made for memory requirements to store the Systems Software modules. The System Software requirements are defined by Task 1.4 (System Software) and are computer system dependent.

The assumed redundancy level for Shuttle subsystems is presented in Table 4-2. In general, these levels are the same as those proposed in Reference 3. In most cases, the basic equations for a redundant subsystem are assumed to occur only once in the coded program. The equations are processed once for each level of redundancy. However, redundant inputs, outputs, and characteristic system parameters are assumed to reside within the computer.

FINAL REPORT

DEVELOPMENT OF SIMULATION COMPUTER COMPLEX SPECIFICATION

MDC E0857
29 JUNE 1973

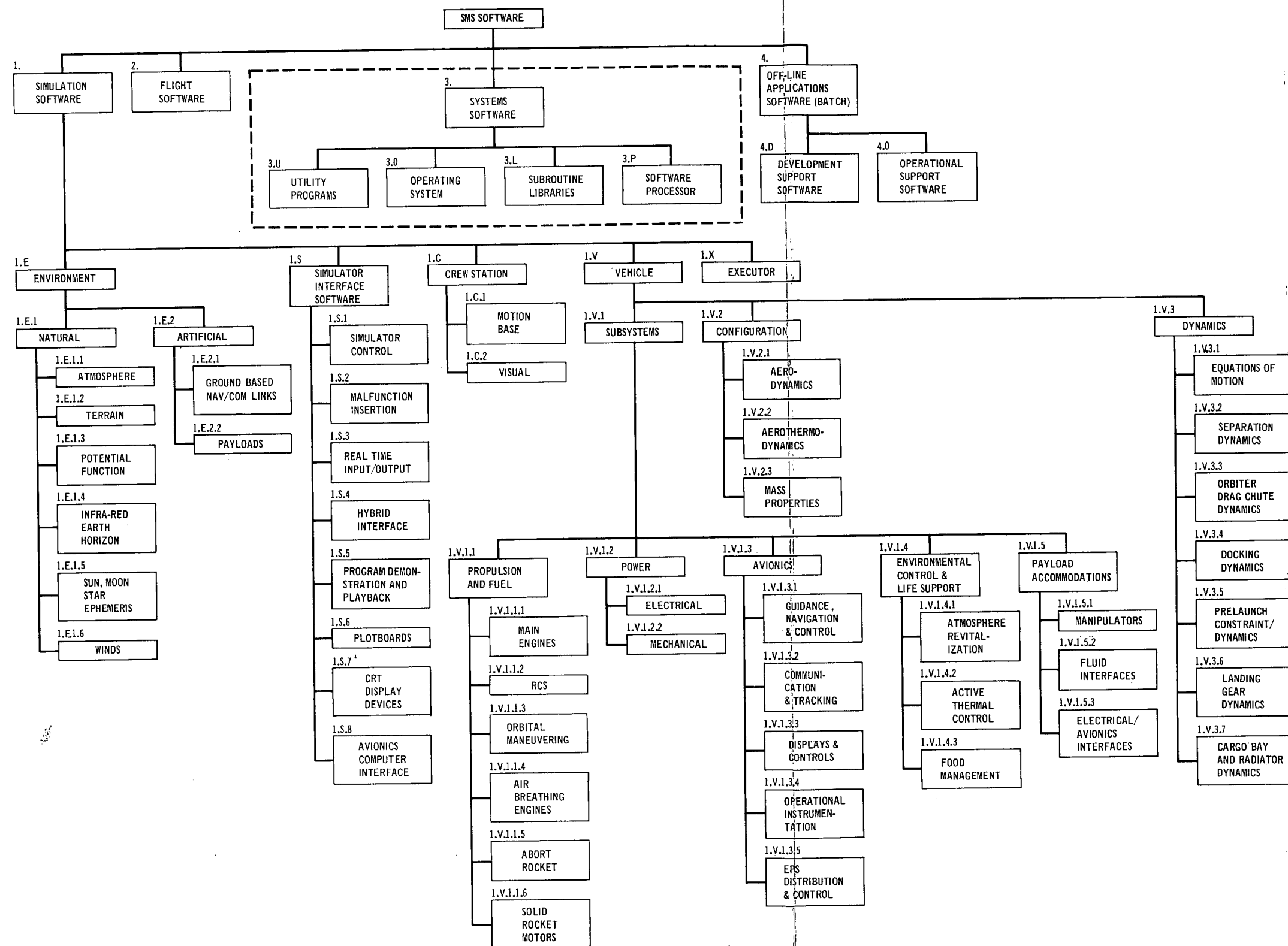


Figure 4-1 Shuttle Mission Simulator Software Module Identification Chart

Table 4-1

SIMULATION SOFTWARE MODULES

MODULE	MODULE FUNCTIONAL DESCRIPTION
1.E.1.1.1 Ascent Atmosphere	1963 Patrick Atmosphere Model. Computations include density, pressure, speed of sound, kinetic and virtual temperature as a function of geocentric altitude. Off-nominal conditions simulated by a percent deviation as a function of altitude.
1.E.1.1.2 Orbital Atmosphere	1971 MSFC Atmosphere Model. Combination of Jacchia's model with Groves Modifications at lower altitudes.
1.E.1.1.3 Entry and Landing Atmosphere	1962 Standard Atmosphere Model. Off-nominal atmosphere conditions implemented by pre-run load of off-nominal data table.
1.E.1.2 Terrain	Provides local terrain elevation data.
1.E.1.3 Potential Function	Model of Earth's potential function including J_2 , J_3 , and J_4 terms. Sun and Moon potential effects on the spacecraft are also included.
1.E.1.4 Infra-Red Earth Horizon	This module determines the altitude of the infra-red horizon above the oblate earth surface. Statistical deviations, generated by an initialization program that runs non-real time are added on to the mean horizon altitude.
1.E.1.5 Sun, Moon, Star Ephemeris	This module computes the vectors describing the Sun and Moon position with respect to the Earth. Daylight/Darkness conditions are computed from these data. Star Ephemeris data is available through a table look-up of star position and brightness.
1.E.1.6 Winds	Wind velocity, gusts, shear, and turbulence effects are simulated as a function of vehicle altitude.

Table 4-1

SIMULATION SOFTWARE MODULES (continued)

MODULE	MODULE FUNCTIONAL DESCRIPTION
1.E.2.1.1 Area Navigation Aids	This module contains search logic and the ground facility station parameters for the navigation systems simulation. The onboard communications and tracking module searches this data file routine for the candidate station parameters. A combination of 500 VOR, DME, TACAN, or ILS ground stations is assumed.
1.E.2.1.2 Space Tracking and Data Network	Ground tracking coverage for the Space Tracking and Data Network System is simulated by this module. Acquisition of Signal (AOS), Loss of Signal (LOS) and measurement angles for each ground station are determined.
1.E.2.2 Payloads	This module simulates a tug vehicle plus one Payload for the tug. A very simplified model of the tug vehicle, including propulsion power, avionics, flight software, dynamics, and mass properties is assumed. A minimal functional simulation of a payload is assumed.
1.S.1.1 Simulation Operating Mode Control	This module provides the instructor/operator with the capability of controlling the simulator operating mode. The moding functions include Operate, Freeze, Reset, Safestore, Write Reset, and Step Ahead.
1.S.1.2 Simulation Control-Buffer for Real Time Data	This module provides for the real time input/output data transfer for simulator control. 400 parameters are assumed necessary for input and/or output.
1.S.2 Malfunction Insertion	This module allows the instructor/operator to insert variable or discrete malfunctions into the simulated systems. It provides the instructor with an efficient means of monitoring and altering the status of the simulated systems. An average of 26 malfunctions is assumed for 36 systems.

Table 4-1

SIMULATION SOFTWARE MODULES (continued)

MODULE	MODULE FUNCTIONAL DESCRIPTION
1.S.3 Real Time Input/Output	This module accepts requests from other simulation software modules requiring input or output to a peripheral device and performs the necessary communication with the Operating system to complete the necessary input/output operations.
1.S.4.1 Hybrid Interface-High Frequency Input	This module performs the hybrid interface operations for the keyboards, hand controllers, and rudder pedals in the crew station. 100 discrete inputs and 68 analog inputs are assumed.
1.S.4.2 Hybrid Interface-Medium Frequency Output	Hybrid interface operations for the visual parameters are performed by this module. 246 Analog and 54 Synchro parameters are assumed.
1.S.4.3 Hybrid Interface-Low Frequency Input/Output	Hybrid interface operations for the crew station switches and display lights are performed by this module. 2500 discrete inputs and 3800 discrete outputs are handled by this module.
1.S.5.1 Program Demonstration and Playback-Control Routine	This module controls the record and playback operations of a simulated flight. When the record function has been selected, the data output from the simulation software modules are recorded on magnetic tape. The playback function inhibits output data from the simulation software modules, and replaces it with data read from the magnetic tape.
1.S.5.2 Program Demonstration and Playback-Buffer	The input and output data transfer for the demonstration and playback module is provided by this buffer data module. 400 parameters are assumed.
1.S.6.1 Plotboards-Drawing Routines	The plotboard system provides a permanent record of the vehicle ground track and radio facilities. This module provides the logic for drawing this data.

Table 4-1
SIMULATION SOFTWARE MODULES (continued)

MODULE		MODULE FUNCTIONAL DESCRIPTION
1.S.6.2	Plotboards-Control Routine	This module checks the switch condition on the recorder and controls the scaling conditions for the pen coordinates for drawing the groundtrack of the vehicle.
1.S.6.3	Plotboards-Recorder Routine	This module checks the switch conditions on the recorder and controls the scaling conditions for the pen coordinates for drawing the vertical profile of the vehicle.
1.S.7	Instructor Console CRT Display Devices	This module processes the request and periodic update of the CRT formats for the 6 instructor CRT devices. The CRT displays are structured such that the text and instructions for supporting each individual display page form a self-contained subroutine which is stored on disk until requested.
1.S.8.1	Avionics Computer Interface-Excluding Manipulators	This module loads and unloads the buffer array containing parameters common to the subsystems modules and the avionics computers. The module loads those parameters not related to the manipulators.
1.S.8.2	Avionics Computer Interface-Manipulators Alone	Loads and unloads the buffer array containing parameters common to the manipulator module and the avionics computers.
1.S.8.3	Avionics Computer Interface-Buffer for Inter Avionics Computer Data	Input/output data buffer array in host computer for inter-avionics computer data transfer.
1.C.1	Crew Station Motion Base	This module provides the drive commands to the motion base system and simulates the vehicle motion during Ascent and Entry.

Table 4-1

SIMULATION SOFTWARE MODULES (continued)

MODULES	MODULE FUNCTIONAL DESCRIPTION
1.C.2.1 Crew Station Visual-Ascent	This module computes the drive signals for the out-the-window visual displays. The forward field of view at the pilot/co-pilot station during the ascent phase includes daylight/darkness, earth scene, and star field scenes.
1.C.2.2 Crew Station Visual-On-Orbit	This module computes the drive signals for the out-the-window visual displays. The forward field of view at the pilot/co-pilot station during the On-Orbit phase includes daylight/darkness, earth scene, star field, and target visual displays.
1.C.2.3 Crew Station Visual-Entry and Aeroflight	This module computes the drive signals for the out-the-window visual displays. The forward field of view at the pilot/co-pilot station during the Entry and Aeroflight phases includes daylight/darkness and either earth scene or terrain map depending on the distance from the landing site.
1.V.1.1.1.1 Main Engine-ON/OFF Equations	This module senses the engine on-off commands for the Main Engine Propulsion System.
1.V.1.1.1.2 Main Engine-Performance and Operations	The Main Engine simulation includes computations of the thrust and fuel system characteristics. This module simulates the operations and performance of the 3 liquid propellant rocket engines plus associated tankage, valves, and plumbing.
1.V.1.1.1.3 Main Engine-Sensor Data	Telemetry sensor data for the 3 Main Engines are maintained by this module. Tank pressures, tank temperatures, fuel and oxidizer weight and quantities, and flow rates will be computed.

Table 4-1

SIMULATION SOFTWARE MODULES (continued)

MODULES	MODULE FUNCTIONAL DESCRIPTION
1.V.1.1.2 Reaction Control Propulsion System	This module simulates the thrust characteristics, pressurization system and propellant system for the reaction control propulsion system. The model includes 40 monopropellant thrusters, 4 pressurization systems, and 3 independent propellant systems.
1.V.1.1.3.1 Orbital Maneuvering System-ON/OFF Equations	This module senses the engine on/off commands for the Orbital Maneuvering Propulsion system and sets appropriate flags.
1.V.1.1.3.2 Orbital Maneuvering System-Performance and Operations	This module simulates the performance and operations of the Orbital Maneuvering Engine and propellant system. Engine thrust, propellant tank, ullage, and associated valve characteristics are modeled.
1.V.1.1.3.3 Orbital Maneuvering System-Sensor Data	Telemetry sensor data for the Orbital Maneuvering Engine and associated propellant system are maintained by this module.
1.V.1.1.4.1 Air Breathing Engine Propulsion System- Performance	This module simulates the operation and performance of the four air breathing engines. The module contains computations for engine thrust, chamber pressures, and temperatures.
1.V.1.1.4.2 Air Breathing Engine Propulsion System- RPM	Engine RPM, sound signals, exhaust gas temperatures, and engine pressure ratios for the four air breathing engines are maintained by this module.
1.V.1.1.4.3 Air Breathing Engine Propulsion System- Engine Oil	This module contains computations of the maximum available horsepower, oil pressure and temperature quantities for the four air breathing engines.

Table 4-1

SIMULATION SOFTWARE MODULES (continued)

MODULE	MODULE FUNCTIONAL DESCRIPTION
1.V.1.1.4.4.1 Air Breathing Engine Propulsion System-Fuel Performance	This module computes the fuel pump conditions, fuel weight, fuel available, fuel tank temperature, and fuel tank pressures.
1.V.1.1.4.4.2 Air Breathing Engine Propulsion System-Fuel Indicators	This routine smooths the analog output for the fuel pressure indicators and fuel temperature indicators.
1.V.1.1.5 Abort Rocket Propulsion	The two abort rockets provide rapid start and the high thrust necessary to separate the vehicle from the booster SRM's and external tank between 0 and 30 seconds from liftoff. The Module senses the abort mode indicator and simulates the performance of the abort rocket system. Thrust, burn rate, case pressure and case temperature are included.
1.V.1.1.6 Solid Rocket Motors	Two Solid Rocket Motors (SRM's) attached to the orbiter external tanks and burning in parallel with the main engine propulsion system provides Ascent propulsive thrust up through staging. This module simulates the performance of the SRM. Altitude corrections for off-nominal thrust are included.
1.V.1.2.1.1 Electrical Power- Battery	The battery module computes the output parameters of current and voltage as a function of the state-of-charge, the charge rate, the charge voltage, power available and power demand.
1.V.1.2.1.2 Electrical Power- Fuel Cells	The cryogenic storage system which provides fuel and oxidizer to the power generating unit (fuel cell) and a model of the fuel cell characteristics and dc power generated are included in this module. Cryogenic flow rates, pressures, ph factors, and efficiency factors are computed during the orbital mission phases.

Table 4-1

SIMULATION SOFTWARE MODULES (continued)

MODULE	MODULE FUNCTIONAL DESCRIPTION
1.V.1.2.1.3 Electrical Power-Generators	This module simulates the operations and performance characteristics of the three 40 Hz generators. Power available and power demand are modeled.
1.V.1.2.1.4 Electrical Power-On-Pad System	The on-pad power module supplies electrical power during the pre-launch phase. Current and voltage supplies and demands are modeled.
1.V.1.2.2.1 Mechanical Power-Auxiliary Power Unit	The APU system provides shaft power to the hydraulic pumps. This module simulates the performance characteristics and operations of the APU system. Exhaust gas temperature, APU supply temperatures and pressures are computed.
1.V.1.2.2.2 Mechanical Power-Hydraulic	This module simulates the hydraulic power system characteristics. The module calculations include system loads, pressures, reservoir quantities and provide power for actuation of the flight controls.
1.V.1.2.2.3 Mechanical Power-Pyrotechnics	Simulates the response characteristics of the pyrotechnics events for manipulator arms ABES valves, fire suppression, drag chute, SRM ignition, abort rocket ignition, separation, and thrust neutralization. Discretes signaling accomplishment or failure of the event and body axis forces and moments due to the explosion are determined by this module.
1.V.1.3.1.1.1 Inertial Measurement Unit	This module simulates the three 4-gimbal platforms with accelerometers. Platform orientation, torquing, equipment errors, telemetry data, temperature control system, and power system demands are included.
1.V.1.3.1.1.2.1 TVC Interface	Simulates the TVC driver electronics (3 units) and the TVC hydraulic pressure monitor (2 units) for the Orbital Maneuvering System.

Table 4-1

SIMULATION SOFTWARE MODULES (continued)

MODULE	MODULE FUNCTIONAL DESCRIPTION
1.V.1.3.1.1.2.2 RCS Interface	Simulates the RCS driver electronics (3 units) and the RCS Thruster Monitor (2 units).
1.V.1.3.1.1.3 Rate Gyros	This module simulates the operations of the triply redundant rate gyros. (9 total). Electrical, thermal, and crew station interfaces are modeled.
1.V.1.3.1.1.4.1 Main Engine Controller	The main engine controller accepts manual commands or commands from the GN&C system and performs the closed loop control of the engine gimbals, throttles, and on/off commands. The controller interfaces with the model of gimbal actuator dynamics. One controller unit is simulated for each engine (3 engines).
1.V.1.3.1.1.4.2 Main Engine Interface	This module performs the data formatting and data selection for the Engine Controller unit. The module is triply redundant for each controller (3 controllers).
1.V.1.3.1.1.5 Horizon Sensor	Three horizon sensors provide data to the GN&C computers for autonomous navigation operations. Simulation dynamics include track, search, and hold modes, errors in the sensed direction due to biases and noise, and simulation of the electrical interfaces.
1.V.1.3.1.1.6 Optical Tracker	This module simulates the triply redundant star/beacon tracker. The module acquires and tracks the brightest object in its field of view and outputs shaft and trunion angles for the GN&C computer.
1.V.1.3.1.2.1 Stability Augmentation Subsystem Electronics	The triply redundant aeroflight stability augmentation system is simulated by this module. The model processes commands from the GN&C computer or the manual controls, and provides stability,

Table 4-1

SIMULATION SOFTWARE MODULES (continued)

MODULE	MODULE FUNCTIONAL DESCRIPTION
	augmentation, and load relief by generating appropriate deflection commands to the control surfaces.
1.V.1.3.1.2.2 Servo Drive and Monitor Electronics	This module simulates for all control surfaces, the operations of the prime servo drive electronics, the prime actuator monitor, the backup servo drive electronics, and the backup actuator monitor. The servo drive electronics conditions signals from the stability augmentation system and generates control signals for output to the secondary actuators. The Monitor tracks the response of the 3 secondary actuators and switches out the servo drive electronics associated with a bad secondary servo.
1.V.1.3.1.2.3 Strapdown Rate Gyro	This module simulates the operations of the backup rate gyro package. This strapdown rate gyro provides body rates in the three axes and is not redundant.
1.V.1.3.1.2.4 Strapdown Accelerometers	This module simulates the operations of the backup accelerometers, one on each of the three body axis. The appropriate interfaces with power switches, circuit breakers and OI sensors are simulated.
1.V.1.3.1.2.5 Air Data Computer	This module simulates the triply redundant Air Data Computer contained in the aeroflight GN&C system. The module computes and conditions data for output to display meters in addition to simulating the Air Data Computer Characteristics.
1.V.1.3.2.1.1 S-Band Equipment	This module detects the presence of the radio signal, determines which antenna is currently selected, and checks all switches and circuit breaker affecting the S-Band system and responds accordingly.

Table 4-1

SIMULATION SOFTWARE MODULES (continued)

MODULE	MODULE FUNCTIONAL DESCRIPTION
1.V.1.3.2.1.2 S-Band Antenna	The Antenna operations of the S-Band communication system is modeled by this module. Communications blackout, occultation of the radio signal, signal strength, antenna gain, and data drop out of the four S-Band antennas is simulated.
1.V.1.3.2.2.1 VHF Equipment	This module tests all switches, circuit breakers, and power signals to determine the mode and operational status of the VHF system.
1.V.1.3.2.2.2 VHF Antennas	Antenna patterns for each of the four VHF Antennas are simulated by this module. Software is included to determine communication blackout, occultation, signal strength, antenna gain, and data dropout.
1.V.1.3.2.3 TACAN	This module simulates the on-board TACAN unit, the L-Band antenna system, and the associated ground generated signals for DME, VOR, and TACAN. The system is triply redundant.
1.V.1.3.2.4 ATC Transponder	This module computes the on-board and associated ground generated signals for the ATC system. The system is doubly redundant.
1.V.1.3.2.5 ILS Receiver	Simulates the on-board ILS receiver unit, the VHF/UHF antennas, and the associated ground generated signals. The system is triply redundant.
1.V.1.3.2.6 Audio Equipment	This module controls the voice communications between the crewmen and ground. The module simulates the functions of the audio center equipment box.

Table 4-1

SIMULATION SOFTWARE MODULES (continued)

MODULE	MODULE FUNCTIONAL DESCRIPTION
1.V.1.3.2.7 Radar Altimeter	The triply redundant radar altimeter, the 6 C-Band antennas, and the associated display and control panel interfaces are simulated.
1.V.1.3.3.1 Timers	This module maintains the value of time for the different display timing systems (event timer, GMT timer, and mission elapsed timer). Logic and computations to support the controls associated with these timers is included.
1.V.1.3.3.2 Artificial Feel System	Rudder pedal and side arm controller force feel signals are generated by this module.
1.V.1.3.4.1 Telemetry	Constructs a simulated PCM telemetry data stream from floating point and discrete parameter values. 3000 parameters are assumed available for telemetry downlink.
1.V.1.3.4.2.1 Caution and Warning-Subsystem Parameter Testing	As part of the double redundant independent caution and warning system, this module performs logical operations on the subsystem sensor data to determine if the data is within tolerances or if redundant systems are in agreement. Warning lights are activated when failures are identified.
1.V.1.3.4.2.2 Caution and Warning-Aural	This module detects failures and annunciates conditions requiring immediate attention of the crew by means of the two aural horns in the crew station.
1.V.1.3.5 EPS Distribution and Control	Distribution, control, and conversion of all electrical power is provided by this module. Switching logic, power conditioning, landing, and distribution are the functions simulated.

Table 4-1

SIMULATION SOFTWARE MODULES (continued)

MODULE	MODULE FUNCTIONAL DESCRIPTION
1.V.1.4.1 Atmosphere Revitalization	Computations for the cryogenic storage quantities, pressures and temperatures, valve response, and for performance for the atmosphere revitalization subsystem of the environmental control system are provided by this module.
1.V.1.4.2 Active Thermal Control	Simulates the external heating and heat exchanger performance and operations.
1.V.1.4.3 Food Management	Power supply demands for the Food Management devices are computed by this module.
1.V.1.5.1.1 Manipulators- Collision Constraints	This module checks the constraints parameters to determine whether or not a manipulator arm is in contact with the orbiter, the payload, or the other manipulator arm.
1.V.1.5.1.2 Manipulators- Subsystem Parameters	Computations for the manipulator subsystem parameter are included in this module. The functional categories of the parameters include: Manipulator/Slave Commands, Manipulator/Slave Motion Sensor outputs, Status Signals, TV Commands, Electrical Power data, and Reference checkout data.
1.V.1.5.1.3 Manipulators- Bending	Flexible body effects for the manipulator arms are simulated. One bending mode per plane for two planes and one torsional mode for each arm section is modeled.
1.V.1.5.1.4 Manipulator Visual System	Computes the drive signals for the out-the-window visual displays at the payload handling station. Two fields of view, the aft looking and overhead views are simulated.

Table 4-1

SIMULATION SOFTWARE MODULES (continued)

MODULE	MODULE FUNCTIONAL DESCRIPTION
1.V.1.5.2 Fluid Interface	Simulates the fluid interfaces between the Orbiter and the payload.
1.V.1.5.3 Electrical/Avionics Interface	Simulates the supply of external power and exchange of avionics data between the Orbiter and the payload.
1.V.2.1.1 Ascent Aerodynamics	Computes the aerodynamic forces and moments during the Ascent mission phase.
1.V.2.1.2 Post Transition Through Landing Aerodynamics	Computes the aerodynamic forces and moments during the post transition through landing and ferry mission phases.
1.V.2.1.3 Orbital Aerodynamics	Computes aerodynamic forces (no moments) during the orbital mission phase.
1.V.2.1.4 Entry Aerodynamics	Computes the aerodynamic forces and moments during the Entry mission phase.
1.V.2.2.1 Aerothermodynamics-Entry	This module computes the orbiter skin temperature due to aerodynamic heating at selected locations on the vehicle. It computes the heat flow rate through the skin into various components (e.g., the aft equipment bay, forward RCS propellant storage area, and eight other areas). This module is used during the entry mission phase.
1.V.2.2.2 Aerothermodynamics-Ascent	During the ascent mission phase this module computes the orbiter skin temperature due to aerodynamic heating at selected locations on the vehicle. Heat flow rates through the skin into various components are computed.
1.V.2.3.1 Mass Properties-Aeroflight and Entry	During the aeroflight and entry mission phases this module computes the vehicle mass properties of gross weight, center of gravity, moments and products of inertia.

Table 4-1

SIMULATION SOFTWARE MODULES (continued)

MODULE	MODULE FUNCTIONAL DESCRIPTION
1.V.2.3.2 Mass Properties-On-Orbit	During the On-orbit mission phase this module determines the vehicle mass properties of gross weight, center of gravity, moments and products of inertia.
1.V.2.3.3 Mass Properties-Ascent	Mass properties for the Ascent mission phase are computed. These include gross weight, center of gravity, moments and products of inertia, and generalized masses for body bending.
1.V.3.1.1 Equations of Motion-Translation	The three degrees of freedom translation equations of motion for the Shuttle and target vehicles are computed by this module. Vehicle acceleration, velocity, position, total velocity, mach no., dynamic pressure, heading and flight path angle are computed.
1.V.3.1.2.1 Main Engine Gimballing	Simulates the response characteristics of the primary and secondary actuators and connecting rod between the actuator for the Main Engine propulsion system.
1.V.3.1.2.2 OMS Engine Gimballing	Simulates the response characteristics of the primary and secondary actuators and connecting rod between the actuator for the Orbital Maneuvering Propulsion.
1.V.3.1.3 Equations of Motion-Rotational	The three degrees of freedom rotational equations of motion for the Shuttle and target vehicles are computed by the module. Vehicle attitude, attitude rate and acceleration, quaternions and quaternion rates, angle-of-attack, sideslip, and bank angle are computed.

Table 4-1

SIMULATION SOFTWARE MODULES (continued)

MODULE	MODULE FUNCTIONAL DESCRIPTION
1.V.3.1.4 Aero Control Surfaces	Simulates the response characteristics for the primary and secondary actuators and the connecting rods for the aero control surfaces. The aero surfaces include: rudder, speed brakes, elevons, and base heat deflector.
1.V.3.1.5.1 Bending and Sloshing-Ascent	During the Ascent mission phase, this module solves the bending and sloshing equations of motion. Two bending modes for the Shuttle and one bending mode for the payload, each including the pitch and yaw plane, are computed. Sloshing modes in two planes are computed for the Shuttle fuel and oxidizer tanks and the payload (tug) fuel and oxidizer tanks.
1.V.3.1.5.2 Bending and Sloshing-Entry	This module solves for the Entry mission phase the generalized bending and sloshing equation of motion. Although the computations are similar to the Ascent phase, less complexity is required, and stored data tables for the Entry phase only are included.
1.V.3.1.6 Coordinate Transformation	Transformation matrices from the BRS coordinate system to the earth reference coordinate system, the BRS coordinate system to the local vertical coordinate system, and the BRS coordinate system to the spherical coordinate system are computed and maintained by this module.
1.V.3.2 Separation Dynamics	The relative dynamics of the separating bodies is maintained by this module for the cases of: Abort Rocket separation, SRM separation, Tank and SRM separation (abort condition), and Tank separation.
1.V.3.3 Orbiter Drag Chute Dynamics	This module senses the drag chute deployment commands and computes the aerodynamic drag effects resulting from the drogue chute and main canopy deployment during the landing phase.

Table 4-1

SIMULATION SOFTWARE MODULES (continued)

MODULE	MODULE FUNCTIONAL DESCRIPTION
1.V.3.4 Docking Dynamics	Precontact parameters and disturbing forces resulting from initial contact of the Shuttle and target vehicle are computed. Commands to switch from separate to mated vehicle configuration are set for the mass characteristics, aerodynamics, and equation of motion modules.
1.V.3.5 Pre-launch Constraint Dynamics	This module constrains the launch vehicle from any vertical motion until the proper conditions (50% ME thrust, SRM Ignition, and tip lock release) have been met.
1.V.3.6.1 Landing Gear Deployment Dynamics	Simulates the dynamic characteristics of lowering and raising the landing gear, the characteristics of the landing gear brakes, parking brakes, anti-skid protection, and more wheel steering.
1.V.3.6.2 Body Dynamics Due to Landing	Computes the ground reaction forces and moments resulting from wheel contact with the runway. This module is also used for taxi and takeoff.
1.V.3.7 Cargo Bay Door Dynamics	Simulates the opening and closing of the cargo bay doors, and the combination of the cargo bay doors and radiator. Latching and unlatching, lock and unlock, and dynamic motion of each door segment is included.
1.X Executor	Controls the execution of the simulator software modules. A pyramid type executor is used to control the calling sequence for the varying rates of execution.

Table 4-2
Simulation Software Modules Affected by Redundancy

SIMULATION SOFTWARE MODULE		REDUNDANCY
1.E.2.1.1	AREA NAVIGATION AIDS	500 STATIONS EACH CONTAINING 5 PARAMETERS
1.E.2.1.2	SPACE TRACKING & DATA NETWORK	20 STATIONS EACH CONTAINING 8 PARAMETERS
1.S.2	MALFUNCTION INSERTION	36 SUBSYSTEMS SELECT - 26 MALFUNCTIONS PER SUBSYSTEM
1.S.7	INSTRUCTOR CONSOLE CRT DISPLAY	6 CRT'S
1.V.1.1.1	MAIN ENGINES	3 ENGINES
1.V.1.1.2	REACTION CONTROL PROPULSION SYSTEM	40 THRUSTERS, 4 PRESSURIZATION SYSTEMS, 3 PROPELLANT SYSTEMS, 3 THERMAL CONTROL SYSTEMS, 3 SENSOR MODULES
1.V.1.1.3	ORBITAL MANEUVERING ENGINES	2 ENGINES
1.V.1.1.4	AIR BREATHING ENGINES PROPULSION	4 ENGINES
1.V.1.1.5	ABORT ROCKET PROPULSION	2 ENGINES
1.V.1.1.6	SOLID ROCKET PROPULSION	2 ENGINES
1.V.1.2.1.2	FUEL CELLS	2 HYDROGEN REACTANT STORAGE TANKS, 2 OXYGEN REACTANT STORAGE TANKS, 3 POWER PLANTS
1.V.1.2.2.1	APU MECHANICAL POWER	4 SYSTEMS
1.V.1.2.2.2	HYDRAULIC POWER	4 INDEPENDENT SYSTEMS
1.V.1.2.2.3	PYROTECHNICS	15 DEVICES
1.V.1.3.1.1.1	INERTIAL MEASUREMENT UNIT	3 REDUNDANT UNITS
1.V.1.3.1.1.2	<u>OMS & RCS INTERFACE</u> OMS	3 REDUNDANT TVC ELECTRONIC DEVICES, 2 REDUNDANT PRESSURE MONITOR DEVICES, (PER ENGINE)
	RCS	3 RCS ELECTRONIC DEVICES, 2 THRUSTER MONITOR DEVICES
1.V.1.3.1.1.3	RATE GYROS	3 REDUNDANT SYSTEMS (9 TOTAL)
1.V.1.3.1.1.4	<u>TVC & PROPULSION INTERFACE</u> ENGINE CONTROLLER	1 PER ENGINE (3 ENGINES)
	ENGINE INTERFACE UNIT	3 PER ENGINE (3 ENGINES)
1.V.1.3.1.1.5	HORIZON SENSORS	3 REDUNDANT UNITS
1.V.1.3.1.1.6	OPTICAL TRACKER	3 REDUNDANT UNITS
1.V.1.3.1.2.1	STABILITY AUGMENTATION SUBSYSTEM ELEC	3 REDUNDANT UNITS
1.V.1.3.1.2.2	<u>SERVO DRIVE AND MONITOR ELECTRONICS</u> SERVO DRIVE ELECTRONICS	5 CONTROL SURFACES, 4 PACKAGES PER CONTROL SURFACE
	MONITOR	2 MONITORS PER CONTROL SURFACE
1.V.1.3.1.2.5	AIR DATA COMPUTER	3 REDUNDANT SYSTEMS
1.V.1.3.2.1	S-BAND SYSTEM	4 ANTENNAS, 1 WIDE-BAND, TRANSMITTER, 2 SGLS S-BAND TRANSPONDERS, 2 USB TRANSPONDERS, 2 SIGNAL PROCESSORS, 2 USB DECODERS, 2 SGLB DECODERS, 2 SGLB INTERROGATORS
1.V.1.3.2.2	VHF SYSTEM	4 VHF ANTENNAS, 2 VHF TRANSCEIVERS (AM & FM)
1.V.1.3.2.3	TACAN	3 REDUNDANT SYSTEMS
1.V.1.3.2.4	ATC TRANSPONDER	2 REDUNDANT SYSTEMS
1.V.1.3.2.5	ILS RECEIVER	3 REDUNDANT SYSTEMS
1.V.1.3.2.7	RADAR ALTIMETER	3 REDUNDANT SYSTEMS
1.V.1.3.5	EPS DISTRIBUTION AND CONTROL	3 REDUNDANT SYSTEMS
1.V.3.1.1	TRANSLATIONAL EQUATIONS OF MOTION	2 VEHICLES (SHUTTLE AND TARGET)
1.V.3.1.2	<u>ENGINE GIMBALLING</u> MAIN ENGINES	18 SECONDARY ACTUATORS, 6 LINKAGE SYSTEMS, 12 PRIMARY ACTUATORS
	OMS ENGINES	12 SECONDARY ACTUATORS, 4 LINKAGE SYSTEMS, 8 PRIMARY ACTUATORS
1.V.3.1.3	ROTATIONAL EQUATIONS OF MOTION	2 VEHICLES (SHUTTLE AND TARGET)
1.V.3.1.4	AERO CONTROL SURFACES	5 AERO SURFACES EACH CONSISTING OF: 2 PRIMARY ACTUATORS, 4 SECONDARY ACTUATORS, 1 LINKAGE SYSTEM, 2 HYDRAULIC POWER SOURCES
1.V.3.1.5	BENDING AND SLOSHING	2 PITCH AND 2 YAW BENDING MODES FOR SHUTTLE, 1 PITCH AND 1 YAW BENDING MODE FOR PAYLOAD, 2 SLOSHING MODES FOR SHUTTLE, 2 SLOSHING MODES FOR PAYLOAD
1.V.3.5	PRE-LAUNCH CONSTRAINT DYNAMICS	8 TIP LOCK RELEASE PINS

DEVELOPMENT OF SIMULATION COMPUTER COMPLEX SPECIFICATION

4.3 DATA SOURCES

The approach taken during the simulation software sizing task relied heavily upon the use of data from existing simulators whenever possible. However in some cases no existing simulator data was available for the modules identified. Table 4-3 presents a summary of the number of modules sized versus the source simulation (i.e., existing simulator). Unfortunately, no source simulation software sizing data was available for 28 of the total 114 software modules. The following subsections discuss very briefly the eleven source simulations, and describe the computer and language of each.

Table 4-3
Summary of Modules vs Source Simulation

SOURCE SIMULATION	NO. OF MODULES
FSI/DC-10 SIMULATOR	40
SKYLAB SIMULATOR	10
BANDITO SIMULATION	4
COMMAND MODULE PROCEDURES SIMULATOR	12
COMMAND MODULE SIMULATOR	7
CHRYSLER LAUNCH VEHICLE SIMULATION	3
SHUTTLE ASCENT ABORT CRITERIA SIMULATOR	2
SHUTTLE DOCKING SIMULATOR	1
SHUTTLE MISSION ENGINEERING SIMULATOR	6
LAUNCH ERROR ANALYSIS PROGRAM	1
E&D MANIPULATOR SIMULATOR	1
NONE	28

4.3.1 Flight Safety Incorporated (FSI) DC-10 Simulator

This simulator is a large, very sophisticated simulation of the DC-10 aircraft. Its applications range from pilot training to engineering evaluation of the aircraft systems performance, and therefore contain detailed models of the aircraft subsystems. The FSI DC-10 Simulator uses assembly language programming on the DDP 124 computer. In general this simulator was used as the data source for the SMS modules characteristic of a commercial aircraft (i.e., Air Breathing Engines, Aeroflight GN&C Components, Communication Subsystems, and Aeroflight Vehicle Dynamics).

4.3.2 Skylab Simulator (SLS)

The Skylab Simulator is used for Flight Crew Training of the Skylab astronauts. The simulation uses a combination of assembly language programming on the DDP-224 computer and FORTRAN IV, H level, programming language on the IBM-360 Model 65 computer. This simulator was used primarily as the data source for the SMS modules characteristic of a spacecraft. (i.e., Spaceflight GN&C Components, Environmental Control and Life Support Subsystems, and EPS Distribution and Control Subsystems).

4.3.3 BANDITO Simulation

BANDITO (Burn and Navigation Dispersions in Transfer Orbits) is a digital computer program developed by MDAC for analysis of the Apollo guidance and navigation system during the rendezvous sequence. The program was coded in FORTRAN on the UNIVAC 1108 computer. A limited number of modules were sized using BANDITO as the source simulation. Those sized included portions of the natural environment, artificial environment, and coordinate transformation modules.

4.3.4 Command Module Procedures Simulator (CMPS)

The CMPS is used for part task procedures development and training of the Apollo and Skylab flight crews in the command module guidance, navigation, and control operations. The CMPS was coded in FORTRAN IV on the CDC 6400 computer. Hybrid Interface modules, Visual modules, and some Spaceflight GN&C Subsystem modules were sized using the CMPS as the source simulation.

4.3.5 Command Module Simulator (CMS)

The CMS is used for flight crew training of the Apollo and Skylab astronauts. The CMS uses assembly language programming on the DDP224 computer. The CMS was used as the data source for the spaceflight propulsion systems (i.e., Main Engine modules, and Orbital Maneuvering modules) and the Inertial Measurement Unit module.

4.3.6 Chrysler Launch Vehicle Simulation (CLVS)

The Chrysler Launch Vehicle Simulation was used as the data source for the flexible body dynamic equations (i.e., Vehicle Bending, Propellant Sloshing). This simulation is a very sophisticated analysis program of the Saturn launch vehicle. The programs were developed on the UNIVAC 1108 computer and uses FORTRAN V programming language.

4.3.7 Shuttle Ascent Abort Criteria Simulator (SACS)

This simulator was a part task, real-time simulator developed to examine the manual control operations during the Ascent phase of the Shuttle Mission and was originally implemented on the CDC 3200 computer. Since only a small section of the total program (prelaunch constraint equations) was of interest, this section of code was recompiled on the CDC 6400 computer. The SACS prelaunch constraint equations were written in FORTRAN IV language.

4.3.8 Shuttle Docking Simulator (SDS)

This simulator was a part task, real-time simulator developed to examine the operational technique requirements of the Shuttle docking phase. Although the simulator was originally developed on the CDC 3200 computer, the sub-routines of interest were recompiled using the CDC 6400 computer and the original FORTRAN IV equations. The SDS was used as the data source for those modules relating to the vehicle docking dynamics.

4.3.9 Shuttle Mission Engineering Simulator (SMES)

The SMES is a fully instrumented, two-man crew station capable of simulating on-orbit and reentry flight phases of the Shuttle Mission. The SMES, developed by McDonnell Douglas Astronautics Company during the Phase B Studies, used FORTRAN IV programming on the CDC 6600 computer. The SMES was used as

the data source for the Translation and Rotation Equations of Motion Modules, and the Vehicle Aerodynamic Configuration Module. For the Aerodynamic Modules, the SMES subroutines were recompiled on the CDC 6400 computer.

4.3.10 Launch Error Analysis Program (LEAP)

LEAP is a three degree of freedom digital simulation which contains the necessary guidance logic for defining the Saturn vehicle ascent trajectory. It contains an error analysis capability for evaluating trajectory dispersions from lift-off through earth orbit insertion. FORTRAN IV language was used to program LEAP on the CDC 3200 computer. LEAP was used as the data source for the natural environment potential function module.

4.3.11 E&D Manipulator Simulator

The manipulator collision constraint module was sized using the NASA Engineering and Development Directorate's manipulator visual simulation as the data source. The simulation uses the Xerox Data Systems Sigma 5 computer.

DEVELOPMENT OF SIMULATION COMPUTER COMPLEX SPECIFICATION

4.4 SIMULATION SOFTWARE SIZING ESTIMATE - CONVENTIONAL APPROACH

Shuttle Mission Simulator Simulation Software module sizing estimates, assuming conventional/simulation techniques, are presented in Table 4-4. Detailed analyses of the estimates for each module may be found in Reference 2. The column headings are defined as follows:

Execution Rate - The number of times the module is executed per second.

Total Instructions - The total number of assembly language instructions stored in the module excluding library routines or utility programs that may be called by the module.

Maximum Instructions Executed - The maximum number of instructions executed (including utility programs) for the longest path through the software (this primarily accounts for branching and do-loops).

Arithmetic Instructions Executed - The number of executed instructions that are floating point arithmetic operations such as "add", "multiply", etc.

Local Variables and Constants - These data consist of a tabulation of the data words within a routine which do not appear in a COMMON block and hence, do not communicate with other routines. Buffer data arrays were assumed to contribute to local variables and constants.

Contribution to Data Base - A module contributes to the data base if it generates parameters which are placed in the COMMON area for communication with other modules.

Million Arithmetic Instructions Per Second - Arithmetic instructions executed multiplied by Execute Rate, divided by one million.

Total Core (Unpacked) - Sum of Total Instructions, Local Variables and Constants and Contribution to Data Base.

Million Instructions Per Second (MIPS) - Maximum Instructions Executed multiplied by Execute Rate and divided by one million.

Mission Phase - Mission Phases in which each module is active. Phases are defined as follows:

- P - Prelaunch - prior to liftoff
- A - Ascent - interval between liftoff and main engine shutdown
- O - On-orbit - interval between main engine shutdown and entry interface (about 400,000 feet altitude)
- E - Entry - interval from entry interface through landing
- F - Ferry - includes horizontal takeoff, cruise, approach and landing

Table 4-4.
Software Module Sizing Estimates – Conventional Approach

MODULE	DATA TYPE	EXECUTE RATE	TOTAL INSTRUCTIONS	MAXIMUM INSTRUCTIONS EXECUTED	ARITHMETIC INSTRUCTIONS EXECUTED	LOCAL VARIABLES AND CONSTANTS	CONTRIBUTION TO DATA BASE	MILLION ARITHMETIC INSTRUCTIONS PER SECOND	TOTAL CORE (UNPACKED)	MILLION INSTRUCTIONS PER SECOND (MIPS)	MISSION PHASE
1.0 SIMULATION SOFTWARE											
1.E ENVIRONMENT											
1.E.1 NATURAL											
1.E.1.1 ATMOSPHERE											
1.E.1.1.1 ASCENT		12	660	1030	104	210	12	0.001	882	0.012	PA
1.E.1.1.2 ORBITAL		12	690	2910	581	177	1	0.007	868	0.035	O
1.E.1.1.3 ENTRY AND LANDING		12	160	300	59	100	2	0.001	262	0.004	EF
1.E.1.2 TERRAIN		12	10	120	9	7230	1	0.000	7241	0.001	EF
1.E.1.3 POTENTIAL FUNCTION		12	290	790	159	26	3	0.002	319	0.009	PAOEF
1.E.1.4 INFRA-RED EARTH HORIZON		1	50	15900	3179	700	3	0.003	753	0.016	O
1.E.1.5 SUN, MOON, STAR EPHEMERIS		12	500	3120	539	65	135	0.001	700	0.003	ACE
1.E.1.6 WINDS		12	160	580	58	155	3	0.001	318	0.007	PAEF
1.E.2 ARTIFICIAL											
1.E.2.1 GROUND BASED NAV/COM LINKS											
1.E.2.1.1 AREA NAVIGATION AIDS		12	90	2200	0	2520	27	0.000	2637	0.026	PAOEF
1.E.2.1.2 SPACE TRACKING AND DATA NETWORK		1	2390	10000	1999	350	42	0.002	2782	0.010	PAOEF
1.E.2.2 PAYLOADS		12	410	1500	149	200	100	0.002	710	0.018	PAOE
1.S SIMULATION INTERFACE SOFTWARE											
1.S.1 SIMULATOR CONTROL											
1.S.1.1 OPERATING MODE CONTROL		1	1150	190	18	510	40	0.000	1700	0.000	PAOEF
1.S.1.2 BUFFER FOR REAL-TIME DATA		1	1200	2000	0	400	0	0.000	1600	0.002	PAOEF
1.S.2 MALFUNCTION INSERTION		6	390	190	18	80	1250	0.000	1720	0.001	PAOEF
1.S.3 REAL TIME INPUT/OUTPUT		143	760	310	0	35	12	0.000	807	0.044	PAOEF
1.S.4 HYBRID INTERFACE											
1.S.4.1 HIGH FREQUENCY INPUT		25	700	1330	132	200	168	0.003	1068	0.033	PAOEF
1.S.4.2 MEDIUM FREQUENCY OUTPUT		12	0	4220	421	0	0	0.005	0	0.051	PAOEF
1.S.4.3 LOW FREQUENCY IN/OUT		6	0	40400	0	0	2500	0.000	2500	0.242	PAOEF

Table 4-4
Software Module Sizing Estimates - Conventional Approach (Continued)

MODULE	EXECUTE RATE	TOTAL INSTRUCTIONS	MAXIMUM INSTRUCTIONS EXECUTED	ARITHMETIC INSTRUCTIONS EXECUTED	LOCAL VARIABLES AND CONSTANTS	CONTRIBUTION TO DATA BASE	MILLION ARITHMETIC INSTRUCTIONS PER SECOND	TOTAL CORE (UNPACKED)	MILLION INSTRUCTIONS PER SECOND	MISSION PHASE
1.S.5 PROGRAM DEMO AND PLAYBACK										
1.S.5.1 CONTROL ROUTINE	12	1450	260	29	25	40	0.000	1515	0.003	PAOEF
1.S.5.2 DEMO AND PLAYBACK BUFFER	12	1200	2000	0	400	0	0.000	1600	0.024	PAOEF
1.S.6 PLOTBOARDS										
1.S.6.1 DRAWING	25	80	180	15	100	50	0.000	230	0.004	AOEF
1.S.6.2 CONTROL	3	660	1520	303	0	0	0.001	660	0.005	AOEF
1.S.6.3 RECORDER	12	290	630	142	0	0	0.002	290	0.008	AOEF
1.S.7 CRT DISPLAY DEVICES	6	4350	3870	386	7118	40	0.002	11508	0.023	PAOEF
1.S.8 AVIONICS COMPUTER INTERFACE										
1.S.8.1 EXCLUDING MANIPULATORS	25	5400	12500	0	1500	0	0.000	6900	0.312	PAOEF
1.S.8.2 MANIPULATORS ALONE	25	990	550	0	340	0	0.000	1330	0.014	0
1.S.8.3 BUFFER FOR INTER AVIONICS COMPUTER DATA	0	0	0	0	3840	0	0.000	3840	0.000	PAOEF
1.C CREW STATION										
1.C.1 MOTION BASE	25	940	1930	384	275	30	0.010	1245	0.048	AEF
1.C.2 VISUAL										
1.C.2.1 ASCENT	12	1570	2670	534	125	35	0.006	1730	0.032	A
1.C.2.2 ON-ORBIT	12	2450	5680	1136	185	45	0.014	2680	0.068	0
1.C.2.3 ENTRY AND AEROFLIGHT	12	900	2460	492	146	15	0.006	1061	0.030	EF
1.V VEHICLE										
1.V.1 SUBSYSTEMS										
1.V.1.1 PROPULSION										
1.V.1.1.1 MAIN ENGINES										
1.V.1.1.1.1 ENGINE ON-OFF	25	20	60	11	3	3	0.000	26	0.001	PA
1.V.1.1.1.2 PERFORMANCE	6	2930	8480	848	183	192	0.005	3305	0.051	PA
1.V.1.1.1.3 SENSOR DATA	3	890	2670	266	65	252	0.001	1207	0.008	PA
1.V.1.1.2 RCS	6	2320	8340	1384	135	348	0.008	2803	0.050	POE
1.V.1.1.3 ORBITAL MANEUVERING										

Table 4-4
Software Module Sizing Estimates - Conventional Approach (Continued)

MODULE DATA TYPE		EXECUTE RATE	TOTAL INSTRUCTIONS	MAXIMUM INSTRUCTIONS EXECUTED	ARITHMETIC INSTRUCTIONS EXECUTED	LOCAL VARIABLES AND CONSTANTS	CONTRIBUTION TO DATA BASE	MILLION ARITHMETIC INSTRUCTIONS PER SECOND	TOTAL CORE (UNPACKED)	MILLION INSTRUCTIONS PER SECOND (MIPS)	MISSION PHASE
1.V.1.1.3.1 ENGINE ON-OFF		25	20	40	7	3	2	0.000	25	0.001	AO
1.V.1.1.3.2 PERFORMANCE		6	2930	5660	565	183	128	0.003	3241	0.034	AO
1.V.1.1.3.3 SENSOR DATA		3	890	1780	177	65	166	0.001	1121	0.005	AO
1.V.1.1.4 AIR BREATHING ENGINES											
1.V.1.1.4.1 PERFORMANCE		6	2310	27400	5479	1880	100	0.033	4290	0.164	EF
1.V.1.1.4.2 RPM		25	90	550	54	0	0	0.001	90	0.014	EF
1.V.1.1.4.3 ENGINE OIL		1	480	5370	536	0	0	0.001	480	0.005	EF
1.V.1.1.4.4 FUEL											
1.V.1.1.4.4.1 SYSTEM PERFORMANCE		1	1240	1830	182	270	60	0.000	1570	0.002	EF
1.V.1.1.4.4.2 SYSTEM INDICATORS		12	60	150	14	0	0	0.000	60	0.002	EF
1.V.1.1.5 ABORT ROCKET		12	160	820	81	140	8	0.001	308	0.010	PA
1.V.1.1.6 SOLID ROCKET MOTORS		12	160	820	81	140	8	0.001	308	0.010	PA
1.V.1.2 POWER											
1.V.1.2.1 ELECTRICAL											
1.V.1.2.1.1 BATTERIES		12	840	1990	198	175	20	0.002	1035	0.024	PAOEF
1.V.1.2.1.2 FUEL CELLS		12	900	2400	479	125	64	0.006	1089	0.029	PAOE
1.V.1.2.1.3 GENERATORS		12	1690	5520	551	200	50	0.007	1940	0.066	PAOEF
1.V.1.2.1.4 ON-PAD POWER		12	770	770	76	10	10	0.001	790	0.009	P
1.V.1.2.2 MECHANICAL											
1.V.1.2.2.1 APU		6	1120	3830	382	145	150	0.002	1415	0.023	PAOEF
1.V.1.2.2.2 HYDRAULICS		12	810	1620	154	116	20	0.002	946	0.019	PAOEF
1.V.1.2.2.3 PYROTECHNICS		25	150	50	0	90	20	0.000	260	0.001	PAOEF
1.V.1.3 AVIONICS											
1.V.1.3.1 GUIDANCE NAVIGATION AND CONTROL											
1.V.1.3.1.1 SPACEFLIGHT											
1.V.1.3.1.1.1 INERTIAL MEASUREMENT UNIT		25	1840	12400	2479	103	570	0.062	2513	0.310	PAOEF
1.V.1.3.1.1.2 OMS AND RCS INTERFACE											

Table 4-4
Software Module Sizing Estimates - Conventional Approach (Continued)

MODULE \ DATA TYPE	EXECUTE RATE	TOTAL INSTRUCTIONS	MAXIMUM INSTRUCTIONS EXECUTED	ARITHMETIC INSTRUCTIONS EXECUTED	LOCAL VARIABLES AND CONSTANTS	CONTRIBUTION TO DATA BASE	MILLION ARITHMETIC INSTRUCTIONS PER SECOND	TOTAL CORE (UNPACKED)	MILLION INSTRUCTIONS PER SECOND	MISSION PHASE
1.V.1.3.1.1.2.1 TVC	25	1190	9410	941	258	60	0.024	1508	0.235	AC
1.V.1.3.1.1.2.2 RCS	25	150	340	34	6	126	0.001	282	0.008	POE
1.V.1.3.1.1.3 RATE GYRO	25	780	3430	343	20	80	0.009	880	0.086	PAOEF
1.V.1.3.1.1.4 TVC AND PROPULSION INTERFACE										
1.V.1.3.1.1.4.1 CONTROLLER	25	1190	5140	514	211	36	0.013	1437	0.128	PA
1.V.1.3.1.1.4.2 INTERFACE	25	80	2590	258	7	81	0.006	168	0.065	PA
1.V.1.3.1.1.5 HORIZON SENSORS	1	770	2350	235	39	21	0.000	830	0.002	O
1.V.1.3.1.1.6 OPTICAL TRACKER	1	1140	5900	589	92	30	0.001	1262	0.006	O
1.V.1.3.1.2 AEROFLIGHT										
1.V.1.3.1.2.1 STABILITY AUGMENTATION SUB ELEC	25	1150	8230	822	422	54	0.021	1626	0.206	PAOEF
1.V.1.3.1.2.2 SERVO DRIVE AND MONITOR ELEC	25	540	7950	794	535	190	0.020	1265	0.199	PAOEF
1.V.1.3.1.2.3 RATE GYRO	25	450	710	70	42	21	0.002	513	0.018	PAOEF
1.V.1.3.1.2.4 STRAP-DOWN ACCELEROMETERS	25	70	260	52	21	12	0.001	103	0.006	PAOEF
1.V.1.3.1.2.5 AIR DATA COMPUTER	25	740	2450	490	316	68	0.012	1124	0.061	PAOEF
1.V.1.3.2 COMMUNICATIONS AND TRACKING										
1.V.1.3.2.1 S-BAND UNIT										
1.V.1.3.2.1.1 S-BAND EQUIPMENT	6	490	580	57	25	50	0.000	565	0.003	PAOEF
1.V.1.3.2.1.2 S-BAND ANTENNA	6	210	950	189	106	14	0.001	330	0.006	PAOEF
1.V.1.3.2.2 VHF TRANSCEIVER										
1.V.1.3.2.2.1 VHF EQUIPMENT	6	250	290	28	25	30	0.000	305	0.002	PAOEF
1.V.1.3.2.2.2 VHF ANTENNA	6	210	950	189	106	14	0.001	330	0.006	PAOEF
1.V.1.3.2.3 TACAN	12	850	3410	337	120	25	0.004	995	0.041	PAOEF
1.V.1.3.2.4 ATC TRANSPONDER	6	80	150	14	6	10	0.000	96	0.001	PAOEF
1.V.1.3.2.5 ILS RECEIVER	12	1140	5400	1079	200	50	0.013	1390	0.065	PAOEF
1.V.1.3.2.6 AUDIO EQUIPMENT	1	500	500	49	10	10	0.000	520	0.000	PAOEF
1.V.1.3.2.7 RADAR ALTIMETER	12	230	790	79	75	18	0.001	323	0.009	PAOEF
1.V.1.3.3 DISPLAYS AND CONTROLS										

Table 4-4
Software Module Sizing Estimates - Conventional Approach (Continued)

MODULE	EXECUTE RATE	TOTAL INSTRUCTIONS	MAXIMUM INSTRUCTIONS EXECUTED	ARITHMETIC INSTRUCTIONS EXECUTED	LOCAL VARIABLES AND CONSTANTS	CONTRIBUTION TO DATA BASE	MILLION ARITHMETIC INSTRUCTIONS PER SECOND	TOTAL CORE (UNPACKED)	MILLION INSTRUCTIONS PER SECOND	MISSION PHASE
1.V.1.3.3.1 TIMERS	25	220	880	87	6	4	0.002	230	0.022	PAOEF
1.V.1.3.3.2 ARTIFICIAL FEEL SYSTEM	12	260	1320	131	50	40	0.002	350	0.016	PAOEF
1.V.1.3.4 OPERATIONAL INSTRUMENTATION										
1.V.1.3.4.1 TELEMETRY	25	64800	43200	0	4500	0	0.000	69300	1.080	PAOEF
1.V.1.3.4.2 CAUTION AND WARNING										
1.V.1.3.4.2.1 SUBSYSTEM PARAMETER TESTING	6	2070	11300	0	100	60	0.000	2230	0.068	PAOEF
1.V.1.3.4.2.2 AURAL WARNING	12	550	2340	0	86	2	0.000	638	0.028	PAOEF
1.V.1.3.5 EPS DISTRIBUTION AND CONTROL	1	12200	95000	9500	570	135	0.009	12905	0.095	PAOEF
1.V.1.4 ENVIRONMENTAL CONTROL AND LIFE SUPPORT										
1.V.1.4.1 ATMOSPHERE REVITALIZATION	1	5170	15200	1519	330	90	0.002	5590	0.015	PAOEF
1.V.1.4.2 ACTIVE THERMAL CONTROL	1	7660	14000	2265	278	125	0.002	8063	0.014	PAOEF
1.V.1.4.3 FOOD MANAGEMENT	1	150	80	0	45	30	0.000	225	0.000	OE
1.V.1.5 PAYLOAD ACCOMMODATIONS										
1.V.1.5.1 MANIPULATORS										
1.V.1.5.1.1 COLLISION CONSTRAINTS	25	3420	23500	1591	1000	20	0.040	4440	0.587	O
1.V.1.5.1.2 SUBSYSTEM PARAMETERS	25	5150	37600	3759	200	400	0.094	5750	0.940	O
1.V.1.5.1.3 BENDING	25	4120	36700	7412	320	0	0.185	4440	0.917	O
1.V.1.5.1.4 VISUAL SYSTEM	12	4900	11400	2279	370	90	0.027	5360	0.137	O
1.V.1.5.2 FLUID INTERFACE	1	100	500	49	20	10	0.000	130	0.000	PAOE
1.V.1.5.3 ELECTRICAL/AVIONICS INTERFACE	1	100	200	0	10	10	0.000	120	0.000	PAOE
1.V.2 CONFIGURATION										
1.V.2.1 AERODYNAMICS										
1.V.2.1.1 ASCENT	25	600	1200	239	760	6	0.006	1366	0.030	A
1.V.2.1.2 POST TRANSITION THROUGH LANDING	25	650	1100	219	10400	6	0.005	11056	0.027	EF
1.V.2.1.3 ORBIT	12	100	500	99	650	3	0.001	753	0.006	O
1.V.2.1.4 ENTRY	25	600	1020	199	3640	6	0.005	4246	0.025	E
1.V.2.2 AERO.THERMODYNAMICS										

Table 4-4
Software Module Sizing Estimates - Conventional Approach (Continued)

MODULE	DATA TYPE	EXECUTE RATE	TOTAL INSTRUCTIONS	MAXIMUM INSTRUCTIONS EXECUTED	ARITHMETIC INSTRUCTIONS EXECUTED	LOCAL VARIABLES AND CONSTANTS	CONTRIBUTION TO DATA BASE	MILLION ARITHMETIC INSTRUCTIONS PER SECOND	TOTAL CORE (UNPACKED)	MILLION INSTRUCTIONS PER SECOND	MISSION PHASE
1.V.2.2.1 ENTRY		1	90	10100	1009	1400	30	0.001	1520	0.010	E
1.V.2.2.2 ASCENT		1	90	6830	682	500	30	0.001	620	0.007	A
1.V.2.3 MASS PROPERTIES											
1.V.2.3.1 AEROFLIGHT AND ENTRY		3	420	720	143	115	30	0.000	565	0.002	EF
1.V.2.3.2 ON-ORBIT		3	420	720	143	115	30	0.000	565	0.002	O
1.V.2.3.3 ASCENT		3	460	960	156	275	34	0.000	769	0.003	PA
1.V.3 DYNAMICS											
1.V.3.1 EQUATIONS OF MOTION											
1.V.3.1.1 TRANSLATION		12	630	7630	2291	105	150	0.027	885	0.092	AOEF
1.V.3.1.2 ENGINE GIMBALING											
1.V.3.1.2.1 MAIN ENGINES		12	890	5330	1065	120	120	0.013	1130	0.064	PA
1.V.3.1.2.2 OMS ENGINES		12	890	3550	709	80	80	0.009	1050	0.043	AO
1.V.3.1.3 ROTATION		25	550	4950	1384	70	135	0.035	755	0.124	PAOEF
1.V.3.1.4 AERO CONTROL SURFACES		12	3220	5370	536	80	80	0.006	3380	0.064	PAOEF
1.V.3.1.5 BENDING AND SLOSHING											
1.V.3.1.5.1 ASCENT		25	4120	40300	8059	450	70	0.201	4640	1.007	A
1.V.3.1.5.2 ENTRY		25	2060	20200	4039	225	35	0.101	2320	0.505	EF
1.V.3.1.6 COORDINATE TRANSFORMATION		12	400	3480	695	65	45	0.008	510	0.042	PAOEF
1.V.3.2 SEPARATION DYNAMICS		12	380	690	139	35	30	0.002	445	0.008	AO
1.V.3.3 ORBITER DRAG CHUTE DYNAMICS		12	150	250	24	70	2	0.000	222	0.003	EF
1.V.3.4 DOCKING DYNAMICS		12	530	2220	221	20	15	0.003	565	0.027	O
1.V.3.5 PRELAUNCH CONSTRAINT DYNAMICS		12	160	1300	129	10	3	0.002	173	0.016	PA
1.V.3.6 LANDING GEAR DYNAMICS											
1.V.3.6.1 DEPLOYMENT DYNAMICS		6	750	1420	138	75	25	0.001	850	0.009	EF
1.V.3.6.2 BODY DYNAMICS DUE TO LANDING		12	320	630	125	25	20	0.001	365	0.008	EF
1.V.3.7 CARGO BAY AND RADIATOR DYNAMICS		6	920	6140	613	50	40	0.004	1010	0.037	O
1.X EXECUTOR		25	230	120	0	10	5	0.000	245	0.003	PAOEF

4.5 SIMULATION SOFTWARE SIZING ESTIMATES - ADVANCED TECHNIQUES APPROACH

The estimates prepared in the previous section, 4.4 have assumed conventional simulation techniques in arriving at a conservative estimate for the SMS simulation software. In this section two techniques which have been used only sparingly in the past were applied to the SMS estimates to arrive at a second estimate. These techniques, a sampled-data approach to dynamic systems simulation, and curve-fit methods in data manipulation, are not new concepts but are advanced in the sense that few real-time simulations have used them extensively. They were used where possible in arriving at the Advanced Technique sizing estimates which are presented in Section 4.5.3.

4.5.1 Sampled-Data Methodology for Real-Time Simulators

Sampled-data simulation methods are defined as those methods that use sampled-data control theory, z-transforms or z-transform notation to develop simulations as opposed to the methods that use numerical integration algorithms or finite differences.

Explicit numerical integration algorithms are based on fitting the solutions to a set of ordinary differential equations by truncated Taylor series. The form of such an algorithm is independent of the actual differential equations to be solved.

Implicit numerical integration and some sampled-data methods, such as Tustin's method applied to a linear transfer function, are equivalent to implicit trapazoidal integration. Some z-transform substitution methods have different algorithms for different orders of integration, that is, for different powers of s in the Laplace transform description of a system. In general, these offer no advantage over the simpler Tustin substitution.

The sampled-data methods described here, such as Fowler's method (References 4 to 14) generate a sampled-data system to simulate the behavior of the particular continuous differential equations describing the system under consideration. There is no fixed algorithm for integration. The method is general and applies to time-varying coefficient and non-linear systems. The

solutions to the sampled-data simulation converge to the true solutions as the sampled period approaches zero. However, the solutions remain stable with good fidelity for much larger sampling periods, producing better accuracy than the solutions obtained by standard numerical integration algorithms at the same sample periods. Advantages of the sampled-data method for training simulators are:

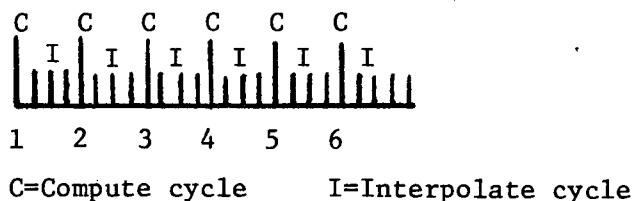
- a) A larger computation step-size can be used without loss of fidelity. This results in less computation time and therefore a lower CPU cost.
- b) Transportation lags inherent in man-in-the-loop simulators can be minimized by making use of transportation leads that are generated by the sampled-data method.
- c) The integration of software and hardware is simplified because the effects of transportation lags resulting from instrumentation, man-in-the-loop, and system hardware in closed loops are evaluated as part of the software development.

Fowler's method has been used for several large-scale simulation problems. One of its first uses was in the development of an all-digital, real-time simulation of Gemini re-entry. This simulation was developed to demonstrate the feasibility of such a simulation and for use in systems studies and guidance computer math flow checkout. Because of the large savings in computer time over conventional simulation, it was also extensively used for control system design studies. This simulation was the first practical, all-digital, six degree of freedom simulation of this size to successfully run faster than real-time on an available digital computer (IBM 7090). The method has been used at MDC for simulating the DC-9 and DC-10 aircrafts, the CMPS Entry simulation, for the Skylab Apollo Telescope Software Simulator, a Shuttle Docking Simulator and a Shuttle Ascent Simulator.

DEVELOPMENT OF SIMULATION COMPUTER COMPLEX SPECIFICATION

Sampled-data methods similar to Fowler's method may be applied to the SMS simulation software to provide reduced software loading estimates for the dynamic systems. In the SMS software, two basic dynamic systems or loops exist, the translational and rotational. These two loops account for a large percentage of the estimated compute load for the host computer. The remaining software modules of the SMS are those on-board vehicle systems which are unrelated to the vehicle dynamics. Of the modules in this group, only the modules pertaining to manipulator arm dynamics have a significant contribution to the compute load. For the purpose of estimating the effect of the sampled-data methods on the SMS Simulation Software, only the modules in the translational and rotational loops, plus those relating to manipulator arm dynamics were considered. Modules falling into these categories cover between 65% and 80% of the total compute load estimated for the various mission phases.

The sample rates of these modules may be reduced by application of the sampled-data method. However, reducing the computation rate does present special problems in interfacing with flight hardware devices which may be in the simulation. These devices will generally require simulated data as inputs at a frequency higher than the sampled-data model execution frequency. This apparent discrepancy between the optimized Simulation Software and the flight hardware requirements can be resolved by employing a technique whereby the required values are provided by using some simple interpolation techniques. In the case of the SMS, where the current flight computer design has a computation rate of 25 samples per second (s/s), if the models in the host machine would be exercised at 6 s/s, then interpolation would occur at 19 s/s. The diagram below indicates a typical one second time frame in the host computer.



The software in the interpolate cycle would be minimal. System configuration (switches, circuit breakers, etc.) would only be checked on the compute cycle.

DEVELOPMENT OF SIMULATION COMPUTER COMPLEX SPECIFICATION

Only parameters required for flight hardware interface or for display equipment (visual displays, meters, etc.) would be generated during the interpolate cycle. The compute cycle of the various models would be distributed to even out the total compute load in the host computer.

The results of applying a sampled-data method such as Fowler's method to the SMS can be summarized as follows:

- a) Dynamic models of the rotational and translational equations of motion and their related modules may be accurately simulated using lower computation rates.
- b) The manipulator bending equations can be simplified using sampled-data methods resulting in a reduction of core requirements. The execution frequency of the manipulator system may also be reduced.

4.5.2 Sophisticated Curve-Fit Methodology for Real-Time Simulations

For a simulation of the Shuttle vehicle, large amounts of data will have to be available to the equations of motion in order to accurately describe the aerodynamic properties of the vehicle. Since the Shuttle will be an entry vehicle and a flying vehicle, even more data will be required than on past simulations. Sophisticated curve-fit techniques offer a method whereby the core required to store the data may be reduced and in some cases the time required to process the data will also be reduced. Some or all of these concepts have been used in each of the real-time and analysis simulations described in References 15 through 17).

4.5.2.1 Transformation of Independent Variables

In the case of aerodynamic data, the basic independent variables are angle-of-attack, α , angle of sideslip, β , and Mach number, M_∞ . Conventional software design techniques use double or even triple table look-up on the aerodynamic force coefficients to provide an accurate simulation. An alternative to this is discussed in Reference 1. In this approach, relative velocity components (u , v , and w) replace angle-of-attack, α , and angle-of-sideslip, β .

The on-line computations, are characterized by the absence of trig terms since u , v , and w are directly obtained from the integration of the equations of motion. The resulting equations give better descriptions of the aerodynamics for large angles of attack than the corresponding linearizations in terms of angle of attack.

4.5.2.2 Curve Fitting by Use of Orthonormal Tables

Curve fitting, as described in Reference 1, can be used in the case where a large number of parameters all depend on a single parameter as in the aerodynamic coefficients or the mass properties. The utility of the scheme described in Reference 1 is even more evident in the flexible body case where there is a large number of additional inertia parameters which can be considered as a function of the total mass.

If one can assume that all the parameters in the collection have some common characteristics (such as the trend in the aero coefficient to be constant as M_∞ approaches 0) then many of the parameters can be represented as a linear combination of just a few. The scheme is just this simple. However, in order to reduce the amount of on-line computation, the few parameters selected are "orthonormalized." The orthonormalized parameters are stored in a conventional tabular form and evaluated with conventional table lookup interpolation techniques.

4.5.2.3 Storage Estimates Based Upon Curve-Fit Techniques

The methods described above have been used to prepare a revised estimate for the amount of core required to store the aerodynamic data. In preparing the estimates it was assumed that the data is transferred from a function of α , β , δe and M_∞ to a function of u , v , w , δe and M_∞ . A Taylor series of 10 terms was assumed to give adequate accuracy. Five orthonormal functions were also assumed, stored at the same density (with respect to M_∞) as the original aero data. Since the instruction rate to generate the aero data was not excessive in the conventional approach, the overall effect of the instruction rate savings due to the curve-fit technique was considered negligible. Therefore, no revisions to instruction rate were estimated in this advanced

technique estimate. Table 4-5 summarizes the effects on aerodynamic data storage due to the curve-fit techniques.

Table 4-5
Aerodynamic Data Storage Requirements

MISSION PHASE	NUMBER OF STORED DATA POINTS REQUIRED	
	CONVENTIONAL APPROACH	CURVE-FIT APPROACH
ASCENT	760	525
ENTRY	3,640	1,320
POST TRANSITION AND LANDING	10,400	1,355
ON-ORBIT	650	650

4.5.3 Sizing Estimates - Advanced Techniques Approach

In light of the advanced techniques presented in the previous two sections, the software modules defined for the SMS were reviewed and the sizing estimates changed where applicable. In general the following modifications were made. Execution rates of all modules related to the translational and rotational loops were reduced due to sampled-data techniques. The modules related to manipulator arm dynamics were also given reduced execution rates. In addition, the amount of software required to simulate manipulator arm bending dynamics was reduced assuming that a much simpler model can be devised with a sample-data system.

The amount of data storage required for aerodynamic data was reduced due to the curve-fit techniques described in Section 4.5.2. No reduction in the total amount of software processing required to generate the aero coefficients was made since the compute load for the conventional approach was already at a reasonable level, and any slight reductions would be insignificant. One new module was created for the advanced techniques approach. This module, I.X.1 Sampled-Data Interpolation Cycles, estimates the special software required to provide the

incremental outputs to the flight hardware and crew station hardware on the "off-cycles" of computation in the sampled-data approach.

Table 4-6 presents the advanced technique sizing estimates for the SMS software modules using the sampled-data and curve fit techniques described above. The conventional estimates are shown for comparison.

TABLE 4-6
CONVENTIONAL TECHNIQUES/ADVANCED TECHNIQUES
SIZING ESTIMATES

() NUMBERS IN PARENTHESES ARE ESTIMATES FOR ADVANCED TECHNIQUES APPROACH.

DATA TYPE	Prelaunch	Ascent	On-Orbit	Entry	Ferry
INSTRUCTIONS STORED	143,350 (146,150)	155,890 (158,690)	169,080 (170,160)	148,520 (151,320)	143,200 (146,000)
LOCALS AND CONSTANTS	28,030 (28,030)	30,890 (30,650)	31,700 (31,700)	52,820 (41,460)	47,180 (38,130)
COMMON	7,740 (7,740)	8,230 (8,230)	8,510 (8,510)	7,710 (7,710)	6,850 (6,850)
TOTAL WORDS	179,120 (181,920)	195,000 (197,560)	209,290 (210,360)	209,050 (200,480)	197,230 (190,980)
FLOATING POINT ARITHMETIC MIPS	0.303 (0.158)	0.586 (0.240)	0.706 (0.222)	0.457 (0.206)	0.434 (0.189)
TOTAL MIPS	3.997 (2.973)	5.491 (3.448)	6.845 (3.511)	4.590 (3.141)	4.445 (3.036)

4.6 SHUTTLE FLIGHT SOFTWARE - SIZING AND IMPLEMENTATION

The purpose of this section is to summarize the basic Shuttle Flight software sizing and timing estimates which were used to identify the host computer memory and computer loading requirements. Estimates of the SMS computer requirements for simulating the flight computer(s) in an Interpretive Computer Simulation (ICS) or a functional simulation mode are presented in Sections 4.6.3 and 4.6.4, respectively.

The Shuttle flight software size was estimated by TRW under an Air Force Contract, "Space Transportation System (STS) Software Concepts Development Study" (Reference 18). The software functions required for Shuttle operation were defined at an application module level. Math flow charts of the modules were established which were used for estimating each module size and timing. The results of the STS study were used as the basis for the sizing estimates during this task.

The STS study technique used to estimate module size and execution time was based upon assembly language programming for fixed point hardware (with scaling required). It was assumed that use of a Higher Order Language (HOL) and floating point hardware would result in equivalent sizing and timing estimates since the additional memory and timing for the HOL generated floating point instructions is offset by the additional fixed point scaling operations required for assembly language generated code.

The Shuttle flight software sizing estimates were structured to represent the loading which would result from the avionics configuration defined in Reference 19. The conclusions reached in this section are not considered very sensitive to changes in this avionics configuration.

4.6.1 Methodology

The flight software modules, were sized in the following manner. For functions identical to those for which sizing and timing data were available from the STS study, the STS estimates were used. For modules that perform different functions from the STS study definition, estimates of sizing and timing were made by comparing with modules that perform similar functions. Modules, for which no similarity with STS modules could be established, were sized by engineering judgement.

The program developed for the STS study, which establishes and maintains a module sizing data base, was modified for this study to perform the following functions:

- (1) To determine the average synchronous task computer loading.
- (2) To assign each 25, 12, 6 and 3 times per second synchronous module to the proper minor cycle.
- (3) To spread the 1 time per second synchronous modules over the minor cycles in a manner that results in approximately the average computer loading during each minor cycle.
- (4) To sort by computer program and determine the total program size.
- (5) To sort (as a second pass) by mission phase and determine the synchronous computer loading (per second and per minor cycle) and determine the size of the flight program for those modules executed during each mission phase.

The modified program was then used to generate the flight program sizing and timing data using the module sizing and timing estimates. The output of the program represented only the synchronous task loading. The asynchronous task loading was then added to the synchronous loading. The asynchronous loading was determined by establishing which asynchronous tasks could be executed during each mission phase. A detailed discussion of the results are presented in the following section.

4.6.2 Flight Software Sizing Estimates

Table 4-7 presents a summary of the flight software sizing and computational loading estimates. The estimates are presented as the total program size and the equivalent peak computational loading and program size per mission phase. The worst case computational loading occurs during the on-orbit coast mission phase.

Table 4-7
Summary of Flight Software Sizing and Flight Computer Loading

MISSION PHASE	COMPUTER PROGRAM	GN&C (2 PROGRAMS)	BACKUP GN&C	PERFORMANCE MONITOR	PAYLOAD	TOTAL
	TOTAL PROGRAM SIZE (WORDS)	164,566	25,284	15,900	24,859	230,609
PRELAUNCH	PROGRAM SIZE (WORDS)	84,000	17,500	15,900	11,900	129,300
	EQUIVALENT PEAK LOADING (MIPS)	1.000	0.300	0.070	0.165	1.535
LAUNCH AND ASCENT	PROGRAM SIZE (WORDS)	87,600	17,500	15,900	11,900	132,900
	EQUIVALENT PEAK LOADING (MIPS)	1.000	0.238	0.070	0.165	1.473
ON-ORBIT COAST	PROGRAM SIZE (WORDS)	105,300	17,600	15,900	24,900	163,700
	EQUIVALENT PEAK LOADING (MIPS)	1.000	0.500	0.070	0.465	2.035
ON-ORBIT POWERED	PROGRAM SIZE (WORDS)	85,300	17,300	15,900	11,900	130,400
	EQUIVALENT PEAK LOADING (MIPS)	1.000	0.238	0.070	0.165	1.473
REENTRY	PROGRAM SIZE (WORDS)	85,400	20,300	15,900	11,900	133,500
	EQUIVALENT PEAK LOADING (MIPS)	0.725	0.195	0.070	0.165	1.155
ENERGY MANAGEMENT	PROGRAM SIZE (WORDS)	79,000	17,900	15,900	11,900	124,700
	EQUIVALENT PEAK LOADING (MIPS)	0.658	0.190	0.070	0.165	1.083
APPROACH AND LANDING	PROGRAM SIZE (WORDS)	78,900	17,400	15,900	11,900	124,100
	EQUIVALENT PEAK LOADING (MIPS)	0.610	0.158	0.070	0.165	1.003
FERRY	PROGRAM SIZE (WORDS)	84,700	17,300	15,900	11,900	129,800
	EQUIVALENT PEAK LOADING (MIPS)	0.800	0.168	0.070	0.165	1.203

DEVELOPMENT OF SIMULATION COMPUTER COMPLEX SPECIFICATION

The mission phases for the Flight Software modules are defined as follows:

- Prelaunch - The phase during which mission planning, such as targeting, time line analysis, targeting verification, etc., as well as on the pad operations utilizing operational software, are performed. This phase ends immediately prior to liftoff.
- Launch and Ascent - This phase begins immediately prior to liftoff at some convenient time such as "guidance reference release". The ascent phase ends at completion of orbit insertion ΔV residual removal.
- On-orbit Coast - This phase consists of coast while in orbit. The on-orbit phase begins with a coast phase at the completion of orbiter orbit insertion and ends with the coasting descent of the orbiter into the atmosphere (400,000 feet altitude) at reentry.
- On-orbit Powered - This phase consists of all thrusting maneuvers while in orbit.
- Reentry - Reentry starts at 400,000 feet altitude and ends at transition from high angle of attack to low angle of attack.
- Energy Management - This phase begins at transition from high to low angle of attack and ends at intercept of the approach and landing navigation aid.
- Approach and Landing - The approach phase begins at the intercept of the approach and landing navigation aid and ends at the completion of rollout. This phase includes the flare and decrab maneuvers.
- Ferry - The ferry phase consists of orbiter site-to-site trips in a horizontal aerodynamic flight mode.

The Shuttle flight program is designed to execute program modules as a function of mission phase; therefore, the computer loading must be defined for each mission phase.

The total program size includes the redundancy in flight software consistent with the Shuttle avionics configuration (e.g., two GN&C, one backup GN&C, one performance monitoring and one payload program). The program size includes the total number of instructions and parameters. No attempt has been made to convert this data into memory words (16 and 32 bit) for the flight computer since the computer has not yet been selected. The reader should exercise caution in comparing the sizing data in Table 4-7 with other sources (e.g., NR proposal) since the sizing estimates may not be in the same units.

The program size per mission phase is the size of the modules active during the specific mission phase. The size per mission phase is dependent upon the software partitioning. Some functions (modules) may be performed in more than one program per mission phase, therefore, the partitioning of the software functions can impact the software sizing estimates presented. The assumed software partitioning is considered a conservative definition for estimating the flight software size and timing.

The software partitioning can impact the computational loading for the synchronous tasks. However, the total computational loading (synchronous plus asynchronous) will in most cases (see Table 4-7) saturate the capacity of the hardware. Therefore, the partitioning impacts the computational time allowed for asynchronous tasks, but does not effect the total computational loading estimates presented in this section.

Reference 1 contains the Shuttle flight software modules by mission phase, the TRW estimated memory requirements (in number of instructions plus parameter storage), timing estimates (in number of instructions executed per each module), and the module execution rates (in number of times executed per second).

4.6.3 Sizing Estimates for Interpretive Computer Simulation (ICS) of Flight Computer

This section presents the SMS host computer computational loading and memory requirements for interpretively simulating the flight computer. A conversion factor for applying the flight software computer loading estimates in Section 4.6.2 has been defined to estimate the SMS computer loading requirements.

The SMS memory estimates include the size of the ICS program plus the size of the flight programs.

4.6.3.1 Methodology

An ICS for a Shuttle crew training simulator was defined as a bit-by-bit model of the flight computer registers, memory and I/O functions. The basic computer operations, such as instruction fetching, instruction decoding, and instruction execution (including truncating of data, setting status word bits, incrementing clock, changing program counter, etc.), must be modeled in sufficient detail to allow operation directly on the flight program machine code. It was assumed that the diagnostic capability, typically found in an ICS used for flight software development, will not be required for a crew training facility. However, provision for logging simulation runs, special real-time displays, malfunction generation, simulation mode control and the communication control between the ICS and the environments are required.

For this study it was assumed that the SMS host computer instruction repertoire would not be compatible with the flight computer. This will establish a worst case computational loading requirement.

The ICS program has been sized by taking existing ICS program sizes and execution speeds presented in Table 4-8 and extrapolating that data to define the estimates for an SMS ICS. The host computer to flight computer instruction ratios were determined from the execution rate data by considering the speed of the host computers with respect to the speed of the flight computers. Two exceptions were the AP-1 and the CMS AGC computers for which the host to flight computer instruction ratios were already known.

4.6.3.2 ICS Sizing Estimates

The worst case SMS computer memory and computational loading requirements for implementing an ICS are:

- Memory - 251K words
- Computational Loading - 61.0 MIPS

Table 4-8
ICS Sizing and Timing Data for Existing Simulators

FLIGHT COMPUTER	DATA SOURCE	ICS HOST COMPUTER	ICS PROGRAM SIZE	EXECUTION RATE	HOST/FLIGHT COMPUTER INSTRUCTION RATIO
SKC-2000	TRW	IBM 360/65	180 K BYTES	28-35 SLOWER THAN REAL-TIME	28-35:1
SKC-2000	KEARFOTT	IBM 370/165	180 K BYTES	13-17 SLOWER THAN REAL-TIME	26-34:1
UNIVAC WSC	TRW	CDC 6500	32 K WORDS	30 TIMES SLOWER THAN REAL-TIME	30:1
APOLLO AGC	MIT	IBM 360/75	NOT AVAILABLE	2 TIMES SLOWER THAN REAL-TIME	35:1
AP-1	TRW	UNIVAC 1108	NOT AVAILABLE	NOT AVAILABLE	30:1
CMS AGC	NASA	DDP-224	16K WORDS	10% FASTER THAN REAL TIME	15:1

These requirements are based upon an estimate of the program size for an ICS and a determination of the ratio of host computer instructions required to model the flight computer instructions.

The ICS is estimated to require approximately 20K words of reference computer memory. This consists of 4K words for the instruction interpretive routines (includes I/O instructions) and 16K words for the basic ICS control functions (redundant copies for four flight programs at 4K words each).

The SMS memory requirements above are based upon the size of the ICS (20K) plus the memory required to store all of the flight programs (231K from Table 4-7). The worst case memory estimates assume that each instruction or parameter in the flight software presented in Table 4-7 is stored in one SMS host computer word (no packing).

The host computer instruction ratios (host computer instructions to model a flight computer instruction) in Table 4-8 range from 15:1 to 35:1. The value of 30:1 has been taken as a representative value for estimating the SMS computer loading requirements.

The worst case SMS computer loading is 61.0 million instructions per second. This estimate is based on the equivalent peak loading (from the worst case computational loading of 2.035 MIPS in Table 4-7 times the host computer instruction ratio (30:1).

If the host to flight computer instruction ratio were reduced by using a host computer that is highly compatible with the flight hardware, this loading could be reduced significantly. However, this estimate (61.0 MIPS) is considered a conservative estimate without placing stringent architectural requirements on the host computer.

4.6.4 Sizing Estimates for a Functional Simulation

The purpose of this section is to present an estimate of the SMS memory and computer loading requirements for a functional simulation of the flight software. Also, included are some of the techniques which may be employed in developing a functional simulator and the relative merits of each.

4.6.4.1 Functional Simulation Groundrules and Assumptions

The traditional functional simulation is a pseudo real-time implementation of the flight software developed from the flight program specification document. This development is typically an independent programming effort parallel to the actual flight software coding activity. The resultant programs in either cases are functionally equivalent. In general, a programming change in the flight software may not affect the functional simulation coding; but a specification change will require re-programming.

It is assumed that a HOL will be utilized in coding the functional simulation. The language used is usually that of the compiler available on the host computer. The flight software for Shuttle will also be implemented using an HOL. This offers a new approach to functional simulation of the flight software;

specifically the use of the same compiler and a unique code generator that outputs machine code for execution on the SMS host computer. The primary advantage to this approach is that a change in the flight program specification requires only a re-compilation of the program after the flight software has been modified and will not require a functional re-programming effort. A disadvantage is that the flight software version used for the functional simulation may have errors that would require "patches" for the functional simulation or would result in a delay until the correction is implemented in the flight software. Another disadvantage to this approach is that the SMS code generator may be developed by a different group of people than those that develop the flight computer code generator and this can result in interpretation problems. A secondary advantage to this approach is that the functional simulation would be a direct result of the actual flight software coding and would not be dependent on interpretation of the program specification document.

Another functional simulation approach is through the use of an object code translator. For this approach, a program must be developed that translates flight computer machine code into SMS host computer instructions that are functionally equivalent. The advantages and disadvantages of this approach are similar to the HOL code generator approach.

For any of the three approaches to the functional simulation, the sizing estimates must include the simulation of the flight computer hardware unique I/O features (where no equivalent instructions exist on the SMS host computer). For sizing this element of the functional simulations, it was assumed the mechanization of these hardware features in a functional simulation are equivalent to the mechanization of the same features in an ICS. However, the memory requirements and the impact on the computational loading of simulating the hardware I/O features will be very small as compared with the total functional simulation. Therefore, this has been neglected in the sizing estimates.

DEVELOPMENT OF SIMULATION COMPUTER COMPLEX SPECIFICATION

4.6.4.2 Functional Simulation Sizing Estimate

The SMS computer memory and computational loading requirements for implementing a functional simulation by any of the approaches outlined in Section 4.6.4.1 are:

Memory - 231K words

Computational Loading - 2035 million instructions per second (MIPS)

These estimates are based on the following assumptions: first, all of the flight software modules must be functionally simulated and second, the resultant coding of the functionally simulated modules will be equivalent to the actual flight software coding in size.

The role of the flight software in a crew simulator is well understood from Apollo experience. The functional simulation must provide a means of simulating every function performed by the flight software. The simulation must, therefore, have an equivalent to each of the flight software modules.

For any of the three approaches to implementing a functional simulation described (traditional approach, an HOL code generator for the SMS host computer, and the object code translator) the flight software is coded initially in an HOL. The differences in compilers, that will impact sizing estimates, are in the efficiency of the code generated. Since the compilers do not exist, so that they can actually be compared, it has been assumed that the degree of optimization for the compilers for the flight computer and the SMS host computer will be equivalent.

Since the SMS host computer compiler generated code will be approximately the same size as the actual flight software code and all of the modules are to be implemented, the total memory requirements for storing the functional simulation and the computer loading for execution of the functional modules will be equivalent to the flight software estimates in Table 4-7.

This is a worst case estimate because the SMS host computer probably will have larger word length and a more sophisticated repertoire of instructions, and should allow programming of the functions with greater ease and efficiency than the flight computer. It is possible, also, that once the flight software is designed, certain functions such as self-test and some executive functions may be abbreviated in the simulation.

4.7 BATCH AND INTERACTIVE USER PROCESSING LOADS

In order to complete the estimate of the software load placed upon the host computer, the batch and interactive computing requirements for the computer complex were considered. This computing in general consists of the processing required to develop the simulation, maintain it, and support it during the operational phase.

Batch computer jobs would typically be those jobs which are entered into the computer's job stream via a local or remote device and which are processed to completion by the computer according to preset user directives. Interactive computer jobs are typically those jobs entered into the computer's job stream which are controlled on-line by the user via some communication device such as a keyboard/typewriter or keyboard/CRT set. Both types of jobs may be classified as non-real-time computing tasks since they will not make use of the real-time hybrid clock. Some of the particular non-real-time computing tasks required for the SMS can be placed into either category, but most can be designated as either one type or the other. The following list defines the non-real-time computing tasks which will be required to support the SMS.

<u>Non-Real-Time Computing Tasks</u>	<u>Batch (B) or Interactive (IA)</u>
● SMS Software Development	
- Source Compilation	B
- Binary and Absolute file Updates	B
- Source library updates	B
- Tape Updates	B or IA
- Flight Software Tape Processing	B or IA
- Software Debug	B or IA
● Training Support	
- Post-training data processing (performance data, telemetry data reduction)	B
- Initial Condition preparation	B
- Training run development and selection	B or IA
- Crew Activity Analysis	B or IA
● On-Line Training Support	
- Simulated ground support (RTCC)	B

There are other types of jobs which will be run on the host computer. These are the operational support-type jobs including simulator fault-detection and fault isolation programs, and pretraining simulator verification programs. These jobs have not been sized as part of this background processing load because they are not processed in addition to the real-time simulation job but in place of it. Also, their core and instruction rate requirements are typically less than those of the simulators, and thus are not the driving items in the estimate. So although these types of jobs are vital to a real-time simulation operation, they do not impose requirements on the background processing load, and thus will not be included in its estimate.

The following rationale was used in estimating the batch and interactive computing load. It was first assumed that, although the batch and interactive jobs are different in their manner of execution, they produce the same load on the computer complex. The batch computing required for the development of the Command Module Procedures Simulator (CMPS) at Building 35, MSC, was used as a basis for the estimating. The CMPS development process is similar to that expected for the SMS in that a higher order language (FORTRAN) was used on a modern, general purpose computer (CDC 6400) and full use was made of the many utility features of the operating system. For this reason, it was felt that the manpower history for development and support of the CMPS, when scaled properly, would provide an accurate estimate of the work force required for the SMS. The estimate of background compute load is then obtained by estimating each man's computer requirements.

Analysis of the software development jobs for the CMPS has shown that each job required peak storage for 75,000 instructions with an average CPU utilization of 200 seconds on the 6400. Assuming an average time of 1.2 μ second per instruction, the average job processed 166 million instructions. For the 115 jobs per day, the batch load estimate for SMS development period is 19.1 billion instructions per day. Assuming background batch may be processed over a two shift period (16 hours) the instruction rate is .332 MIPS.

4.8 SUMMATION OF SIZING DATA

The simulation software sizing estimates were presented in terms of each module size in Section 4.4. Sizing data assuming the use of advanced simulation techniques was presented in Section 4.5. Flight software and batch user estimates were presented in Sections 4.6 and 4.7. In this Section these software loadings are combined or added to establish the total software loading for several simulator configurations.

Section 4.8.1 summarizes estimates for Configuration 1 which utilizes actual flight computers or mini-computers which emulate the flight computers. Estimates for Configuration 2, which contains an interpretive simulation of the flight computers in the host computer, are presented in Section 4.8.2. Finally, Section 4.8.3 presents the estimates for Configuration 3, which utilizes a functional simulation of the flight software in the host computer.

4.8.1 Configuration 1 - Simulation Software in Host Computer with Flight Software in Flight Computers

Figures 4-2 and 4-3 contain graphical presentations of the Reference Computer core and MIPS loading by mission phase for a conventional approach to simulator implementation. Figures 4-4 and 4-5 present the same type data for the advanced technique approach. As noted, these figures present the total software loading including batch and interactive user requirements but do not include any allowances for system software or operating system overhead. The latter are dependant on the particular computer being considered and must be assessed as additional loadings in terms of each computer considered.

In all cases, the maximum loading occurs during the on-orbit mission phase largely due to the heavy loading imposed by the payload manipulator simulation. The maximum loadings are 7.2 MIPS and 284,000 memory words with conventional simulation techniques, and 3.8 MIPS and 285,000 memory words with advanced techniques.

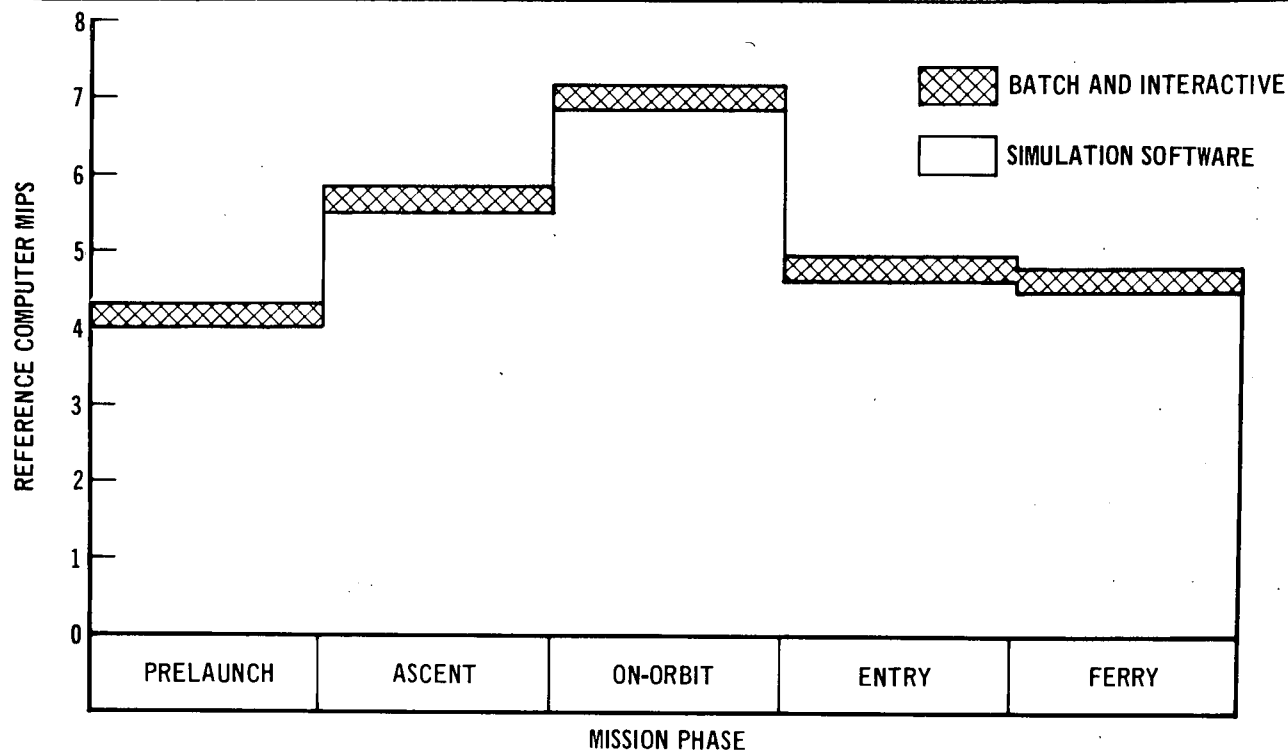


Figure 4-2 Reference Computer MIPS vs Mission Phase:

Conventional Approach - Configuration 1

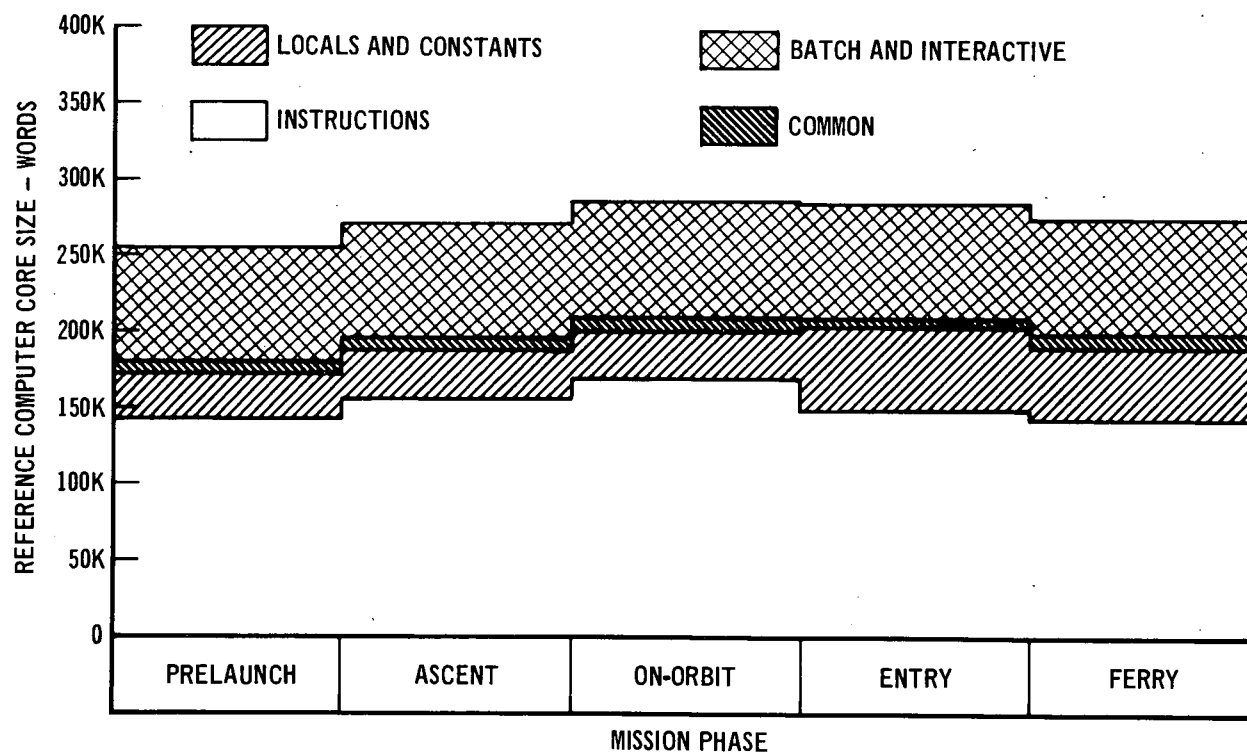


Figure 4-3 Reference Computer Core Size vs Mission Phase:

Conventional Approach - Configuration 1

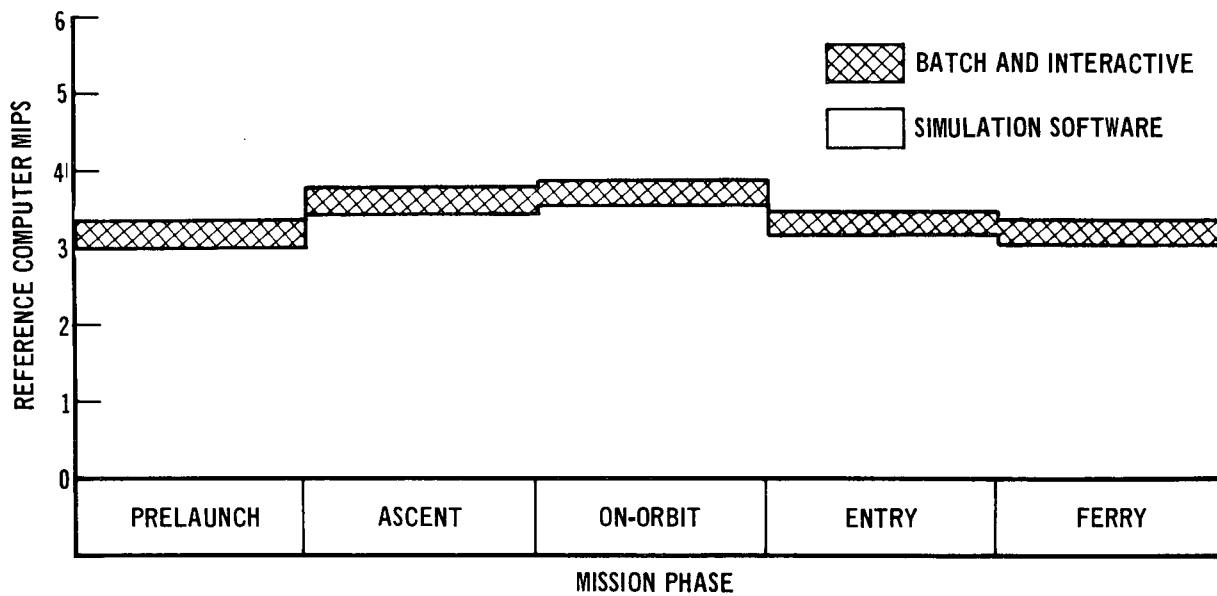


Figure 4-4 Reference Computer MIPS vs Mission Phase:
Advanced Technique Approach – Configuration 1

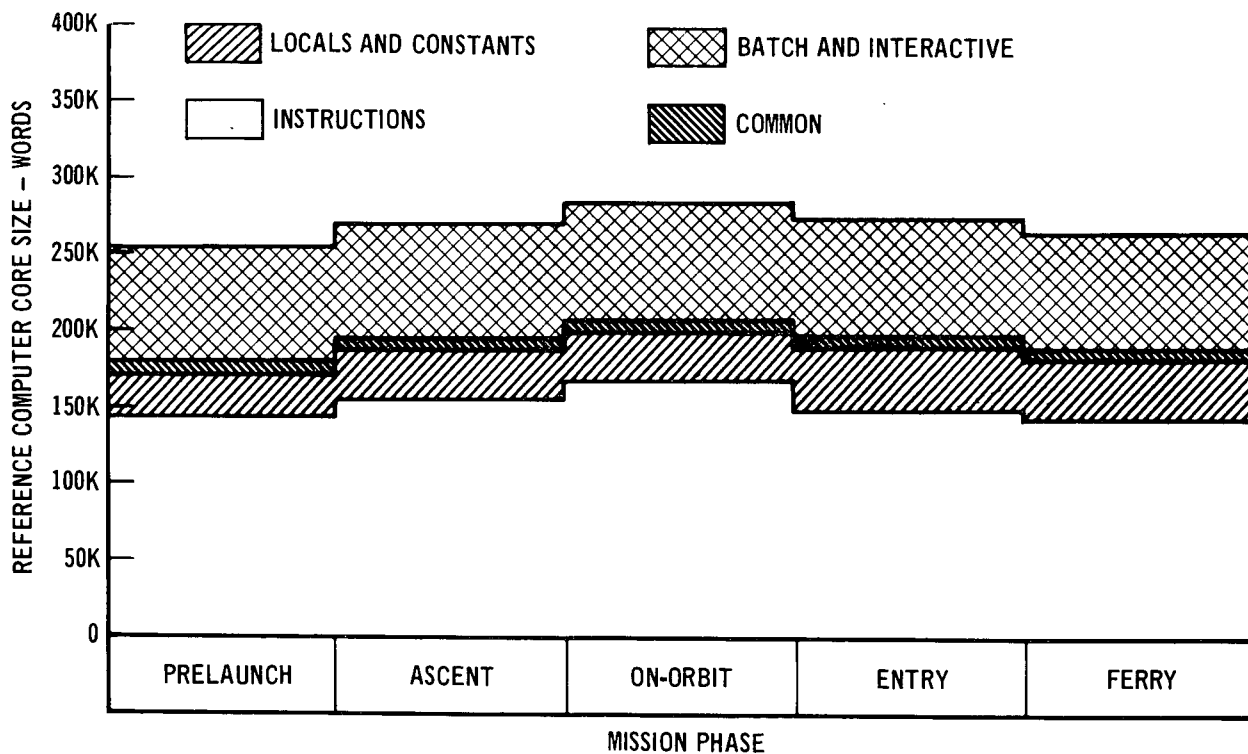


Figure 4-5 Reference Computer Core Size vs Mission Phase:
Advanced Technique Approach – Configuration 1

4.8.2 Configuration 2 - Simulation Software Plus Interpretive Simulation of Flight Computers in Host Computer

The host computer load for interpretively simulating the flight computers was presented in Section 4.6.3. Only data for the on-orbit coast mission phase, which represents worst case loading, were presented. Combining these data with the Simulation Software estimates (209,000 core and 6.8 MIPS) and the batch and interactive user processing load (75,000 core and .3 MIPS) results in Reference Computer core requirements of 535,000 words and 68.2 MIPS. Because of the very large magnitude of these numbers, indicating the seeming infeasibility of this configuration, it seemed inappropriate to reduce and present the data for all mission phases. However, should the user require this data, enough visibility has been provided in other sections to obtain the numbers with a minimal amount of effort.

4.8.3 Configuration 3 - Simulation Software Plus Functional Simulation of the Flight Software in Host Computer

In Section 4.6, sizing estimates for the flight software and its functional implementation were derived. In Figures 4-6 and 4-7, these flight software loadings have been added to the basic simulation and batch loadings. Maximum loadings again occur during the on-orbit mission phase. The maximum MIPS loading is 8.9 and the maximum memory loading is 436,000 words.

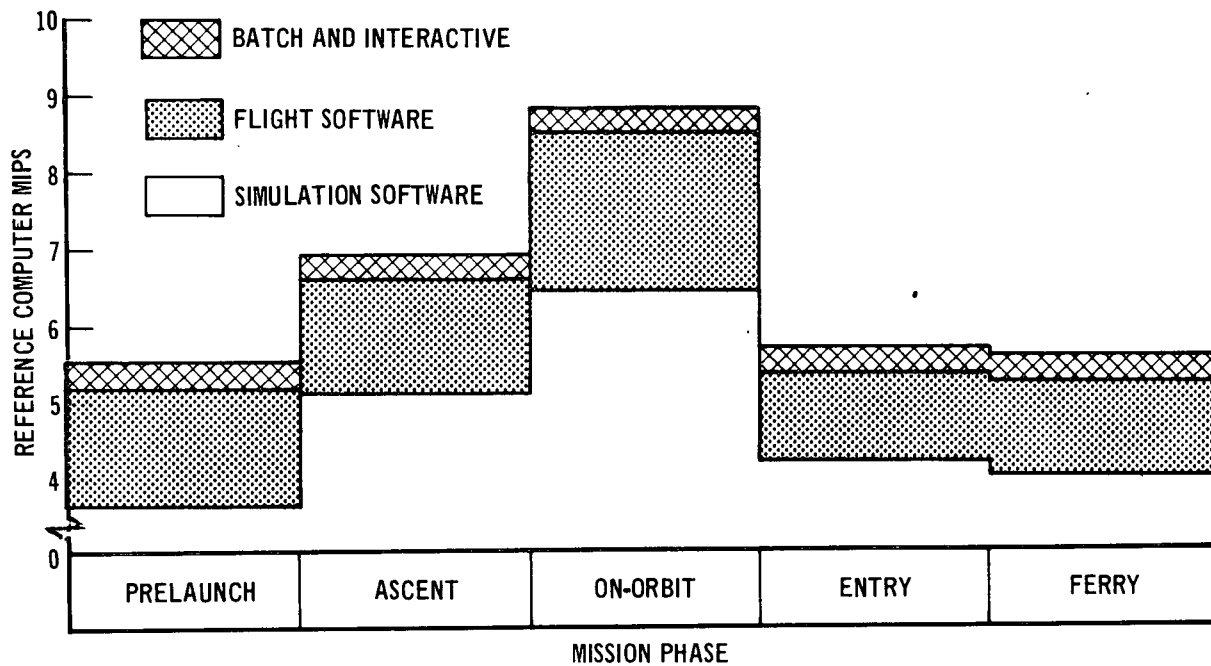


Figure 4-6 Reference Computer MIPS vs Mission Phase:
Conventional Approach - Configuration 3

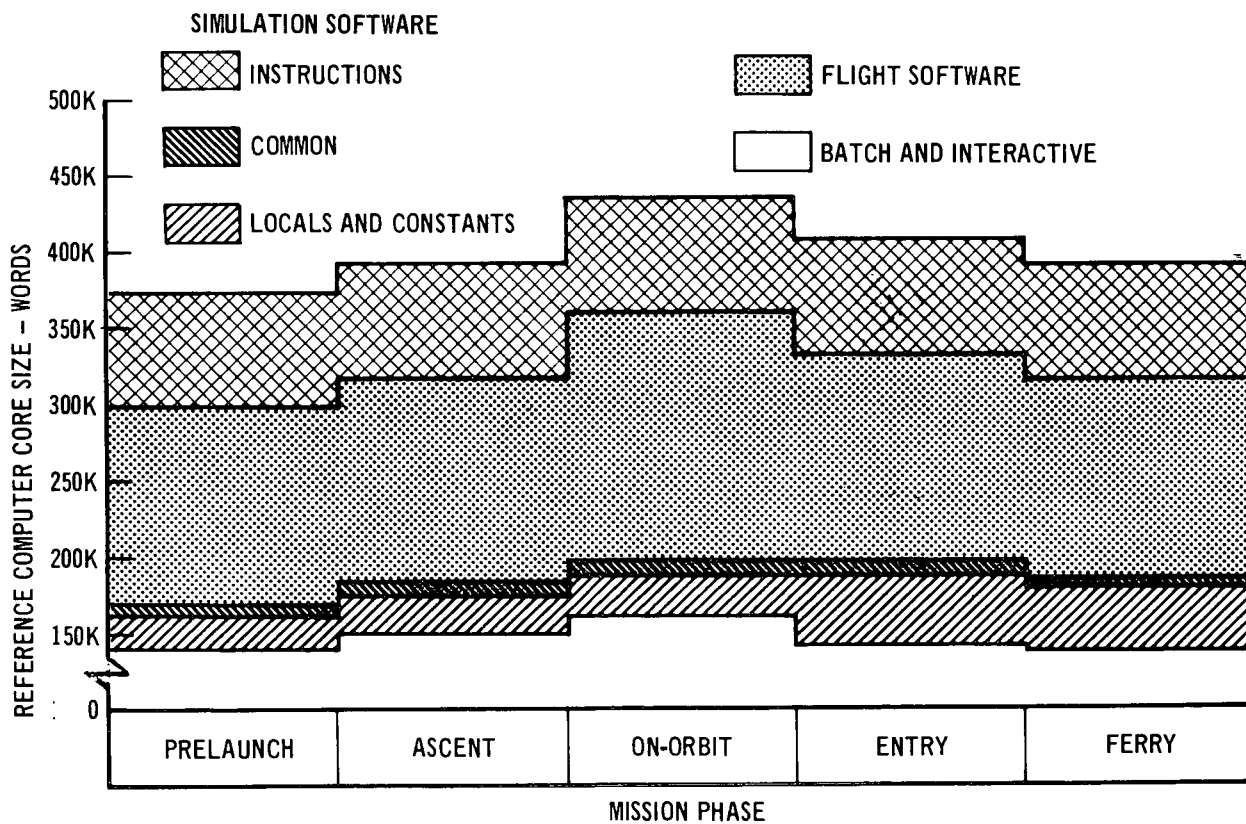


Figure 4-7 Reference Computer Core Size vs Mission Phase:
Conventional Approach - Configuration 3

SECTION 5

COMPUTER HARDWARE REQUIREMENTS - TASK 1.2

This section summarizes the results of the Task performed to determine the computer hardware requirements for implementation of the software loadings discussed in the previous section. The results of this Task have been documented in greater detail in Reference 20.

Training Simulation Computer Complex (TSCC) computer hardware requirements have been established by analysis of the simulation software loadings, by review of simulation facility operational requirements, by review and definition of vendor independent performance criteria, and by computer simulation of the predicted software load. The resulting computer hardware requirements have been summarized and are presented in Section 5.5.

The Shuttle Mission Simulator Simulation Software sizing data presented in the Simulation Software Sizing Report, Reference 1, was further analyzed to establish the requirements for implementation of this software on computers with single and multiple central processors, allowing for factors such as compute load distribution between the 40 millisecond frames expected for the SMS. Memory requirements, the opportunities for software fragmentation into overlays, and the associated input/output requirements have also been analyzed. The results of these software load analyses are summarized in Section 5.2.

The software loading defined in Reference 1 was based on a definition of a Reference computer. The need to specify the processing requirements in a general manner, which would assure adequate performance levels in all proposed candidate computers, required the definition of a performance requirement insensitive to vendor hardware features. This prompted the analysis of characteristic simulation software modules and the development of an SMS operations mix as a computer independent performance requirement. This analysis is presented in Section 5.3.

Implementation of the simulation software load on the baseline, single processor computer configuration was simulated to determine whether any bottlenecks or system inadequacies could be located which were overlooked in development of the simulation software load. The scope of this simulation activity is discussed in Section 5.4.

The summary of the computer hardware requirements and the rationale for their development is presented in Section 5.5. These requirements are based upon the results of the software load analyses, the simulation activity, and experience with both training and engineering development simulation facilities at Johnson Space Center and at the McDonnell Douglas Flight Simulation Laboratory in St. Louis.

5.1 SIMULATION SOFTWARE IMPLEMENTATION

A Simulation Software configuration which is considered to be most representative of the final SMS was recommended in Reference 1 for use in all further TSCC study items. This configuration consisted of the conventional approach to implementing the Simulation Software, with the flight software contained in either flight computers or emulated flight computers. The software load defined by this recommended Simulation Software configuration was used as the basis for defining the computer hardware requirements.

In this Section, the modularity of the Simulation Software, the associated input/output requirements and some timing implications associated with implementation of the software load are reviewed briefly.

5.1.1 CPU Loading

The estimates for the Simulation Software are based upon individual estimates for each of 114 software modules which were defined for the total application software package. Table 5-1 presents summary data for the major module categories and for each of the five mission phases. These data provide visibility to assess the effects of changes in simulation requirements or changes in the assumed simulator configuration. For example, eliminating the need for a payload manipulator simulation in the SMS host computer would lower the instruction processing requirements by approximately 2.6 MIPS for the On-Orbit phase. This change can be reflected into the requirements by either reducing the load on a single CPU configuration or by eliminating a portion of a multiple CPU or multiple computer configuration.

Table 5-1
Shuttle Mission Simulator Software Summary

SOFTWARE MODULE CATEGORIES	PRELAUNCH		ASCENT		ON-ORBIT		ENTRY		FERRY	
	CORE	MIPS	CORE	MIPS	CORE	MIPS	CORE	MIPS	CORE	MIPS
1.E ENVIRONMENT	7,648	.082	8,348	.086	8,769	.117	14,969	.078	13,559	.058
1.S SIMULATOR ENVIRONMENT	34,758	.737	35,938	.753	37,268	.766	35,938	.753	35,938	.753
1.C CREW STATION	-	-	2,975	.080	2,680	.068	2,306	.078	2,306	.078
1.V VEHICLE										
1.V.1 SUBSYSTEMS										
1.V.1.1 PROPULSION	7,957	.130	9,541	.120	7,190	.090	9,293	.237	6,490	.187
1.V.1.2 POWER	7,475	.171	6,685	.162	6,685	.162	6,685	.162	5,596	.133
1.V.1.3 AVIONICS										
1.V.1.3.1 GUIDANCE NAVIGATION AND CONTROL	9,911	1.088	11,137	1.314	11,906	1.138	8,306	.894	8,024	.886
1.V.1.3.2 COMMUNICATIONS AND TRACKING	4,854	.134	4,854	.134	4,854	.134	4,854	.134	4,854	.134
1.V.1.3.3 DISPLAYS AND CONTROLS	580	.038	580	.038	580	.038	580	.038	580	.038
1.V.1.3.4 OPERATIONAL INSTRUMENTATION										
1.V.1.3.4.1 TELEMETRY	69,300	1.080	69,300	1.080	69,300	1.080	69,300	1.080	69,300	1.080
1.V.1.3.4.2 CAUTION AND WARNING	2,868	.096	2,868	.096	2,868	.096	2,868	.096	2,868	.096
1.V.1.3.5 EPS DISTRIBUTION AND CONTROL	12,905	.095	12,905	.095	12,905	.095	12,905	.095	12,905	.095
1.V.1.4 ECLS	13,653	.029	13,653	.029	13,878	.029	13,878	.029	13,653	.029
1.V.1.5 PAYLOAD										
1.V.1.5.1 MANIPULATORS					19,990	2.582			-	-
1.V.1.5.2 FLUID INTERFACE	130	.0005	130	.0005	130	.0005	130	.0005	-	-
1.V.1.5.3 ELECTRICAL/ELECTRONIC INTERFACE	120	.0002	120	.0002	120	.0002	120	.0002	-	-
1.V.2 CONFIGURATION	769	.003	2,755	.040	1,318	.008	17,387	.065	11,621	.029
1.V.3 DYNAMICS	5,948	.310	12,968	1.461	8,600	.437	9,287	.846	9,287	.847
1.X SIMULATION EXECUTIVE	245	.003	245	.003	245	.003	245	.003	245	.003
TOTALS	179,121	3.997	195,002	5.491	209,286	6.845	209,051	4.590	197,226	4.445

Although the total software load is very large, its composition is such that it lends itself to implementation on a variety of computer configurations. In establishing a computer configuration, the modular software approach allows distribution of the load into separate computers or dedicated CPU's. For example, the Payload Manipulator simulation software and the Telemetry portion of the Operational Instrumentation simulation software are two modules which could be easily separated from the remainder of the software load. The Payload Manipulator module is essentially a self-contained simulation of the payload handling system onboard the Shuttle, and could be implemented in a separate computer or a dedicated CPU. This would facilitate concurrent training in different mission phases. The Telemetry software module could be configured in a similar manner since it is, in a sense, an I/O processor, providing an interface between the ground (simulated or real) and the simulated Shuttle. This loading might be placed on already available equipment, and if separated, would again facilitate independent training benefits. Consideration of these and other candidate modules amenable to distribution permits the selection of computer configurations based upon multiple CPU's or multiple computers, in addition to a single large CPU configuration.

5.1.2 Input/Output Loading

Details of the I/O data transfer load were needed as background data for the hardware requirements definition. This subsection summarizes the assumed I/O data transfer between the simulation software and the eleven hardware devices which support the SMS. These eleven hardware devices include:

GN&C Flight Computer No. 1	Mission Control
GN&C Flight Computer No. 2	Simulator Hardware
GN&C Flight Computer No. 3	Instructor Console
Performance Measurement (PM) Flight Computer	Host Computer Tape Device
Payload Handling (PLH) Flight Computer	Host Computer Disc Device
Flight Hardware CRT's	

Supporting data and rationale for the numbers presented are include in the Simulation Software Sizing Report, Reference 1, and the Shuttle Mission Simulator Software Module Sizing Summaries documented in Reference 2. A detailed analysis may be found in Reference 20, the Computer Complex Hardware Requirements Final Report.

Figure 4-1 summarizes the data flow assumptions and provides the description of the input and output data words transferred for the various hardware devices. No direct

FINAL REPORT

DEVELOPMENT OF SIMULATION COMPUTER COMPLEX SPECIFICATION

MDC E0857
29 JUNE 1973

interface between flight computers or flight computers and the flight hardware CRT's will be implemented in the SMS. It is required that all communication to and from the flight hardware pass through the host computer in order that simulated failures may be inserted to provide a realistic training environment for the flight crew.

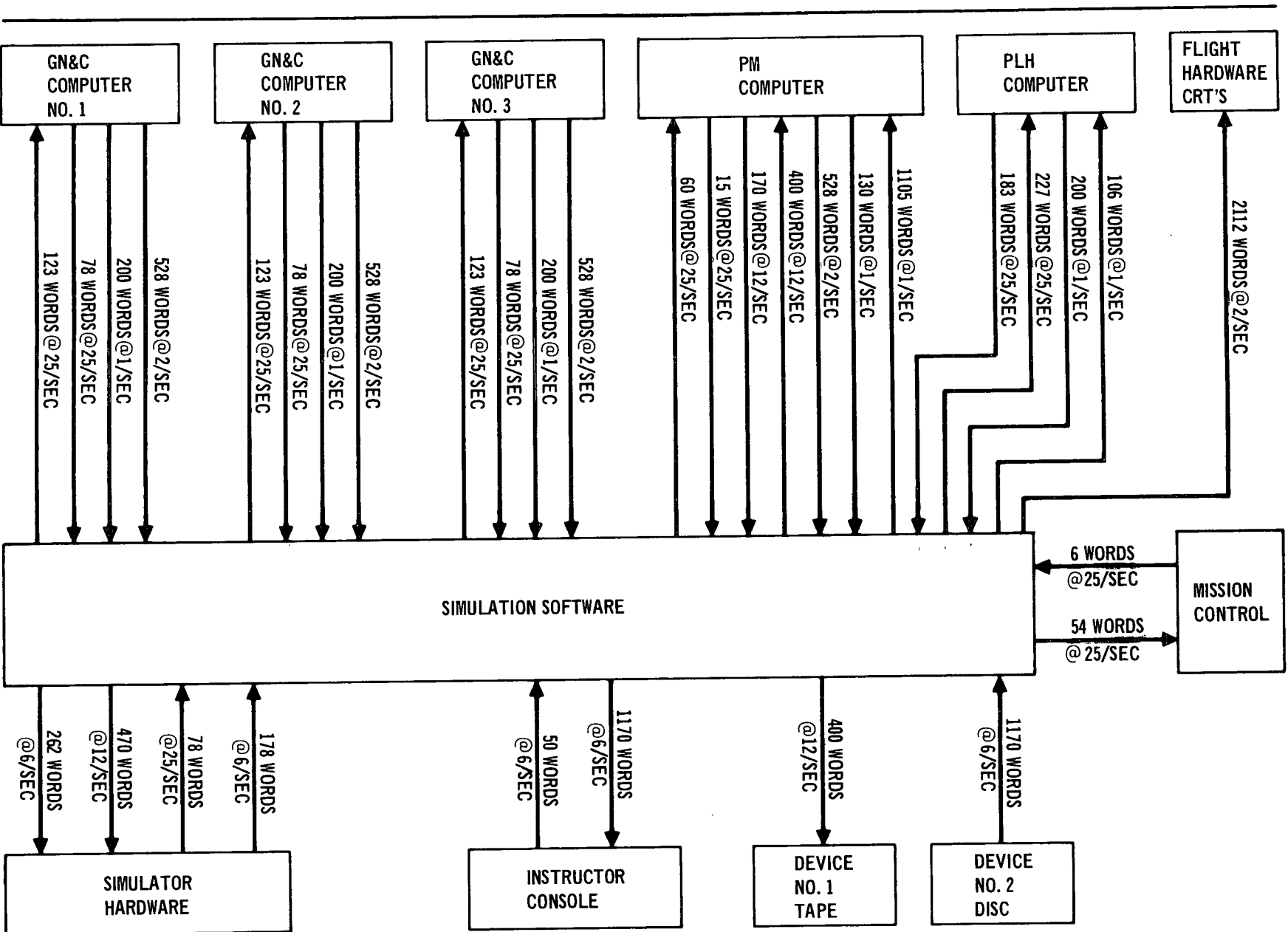


Figure 5-1 I/O Requirements Summary

5.1.3 SMS Computation Process

This section presents a functional description of the SMS computation process. The real-time operational loop assumed for the SMS begins with the host computer in a batch processing state with all simulation I/O completed. Note that for this description, interactive time sharing operation is considered to be part of the general batch loading; since the real-time simulation's priority overrides time-sharing priority for all modes of operation. The first event is a clock interrupt which switches processing in all of the interface minicomputers to the "frame" start routine. These minicomputers then proceed to start the new frame by supervising the sequence of inputs from each minicomputer to the host computer. Near the completion of the input stream, a second clock interrupt signals the host computer to begin processing the simulation problem. At this time the host computer suspends batch processing and re-initiates the suspended simulation program. The second clock interrupt is based upon a delta time which is programmed by the user. The simulation program completes all required processing for the simulation frame in progress, initiates all required I/O operations to the simulation equipment, and notifies the host computer operating system that it can resume the previously suspended batch processing.

The batch processing then continues until the clock interrupt again signals that the inputs from the minicomputer have been completed and the simulation job can begin its next frame. An exception occurs if that portion of the system designated to tracking the real-time operation determines that there is insufficient time to process all output prior to the next clock interrupt. In this instance the simulation program is notified and may take appropriate action, which in most instances should result in termination of the job.

Figure 5-2 presents a functional timeline description of the SMS computation process in terms of the simulation equipment and central processor activities of the host computer. The execution periods for the batch and Simulation Software loads are shown. The period of time between the first clock interrupt and the next is defined as a frame. For the SMS, it is required that 25 frames be processed every second. Therefore, the time period for a frame is 40 m sec. This frame time is required at 25 frames per second because of the necessity of the host computer to interface with the Flight Computers which are being processed.

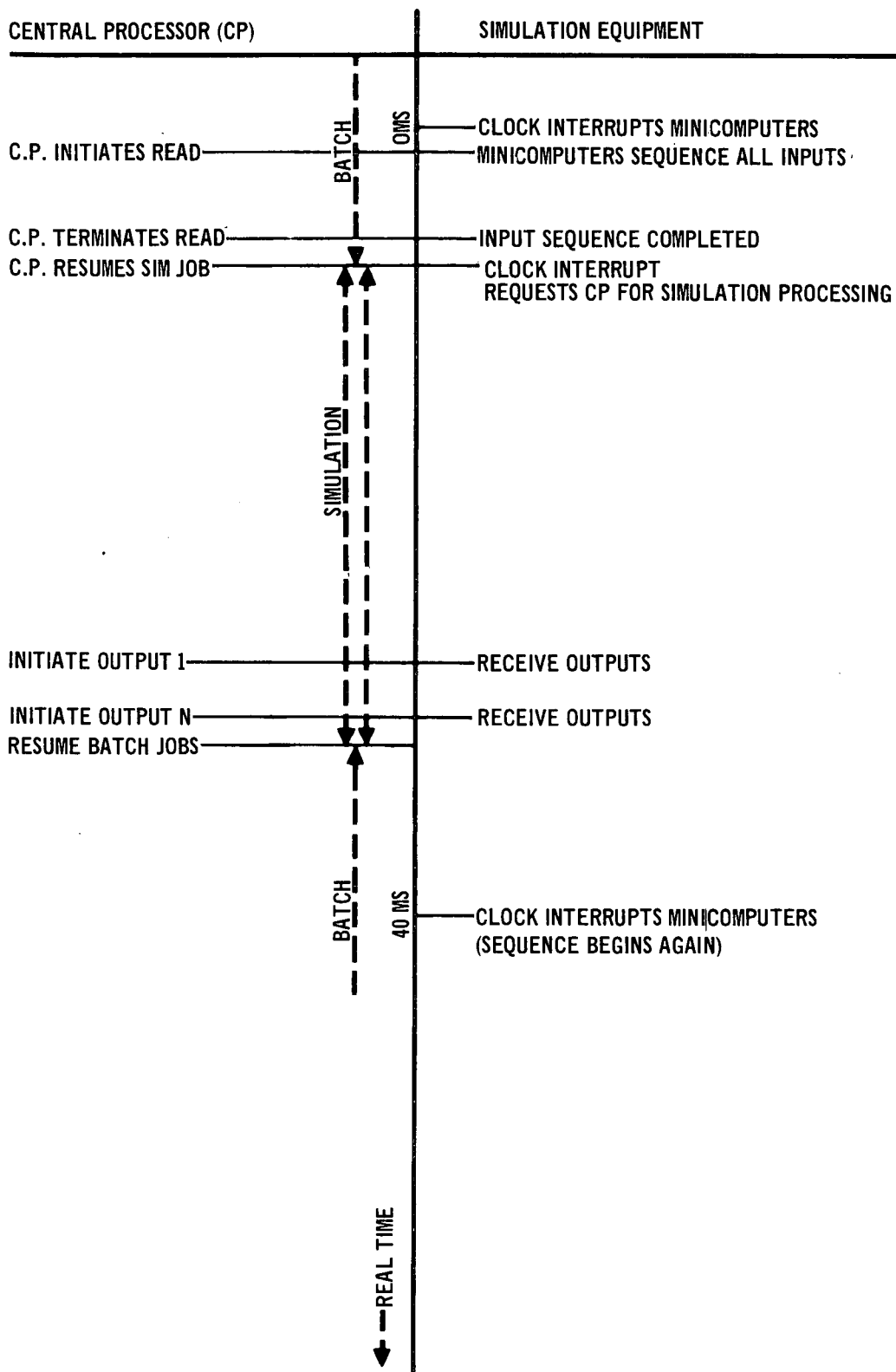


Figure 5-2 Functional Timeline of SMS Computation Process

DEVELOPMENT OF SIMULATION COMPUTER COMPLEX SPECIFICATION

5.2 SIMULATION SOFTWARE LOAD ANALYSIS

In order to realistically state the computer complex hardware requirements, a detailed analysis of the real-time Simulation Software and non-real-time software loads was performed. Because it seemed likely that some vendors may select a computer configuration other than a single large CPU, two additional configurations were included in the analyses. A dual and a four CPU configuration were selected for this analysis since their characteristics were most representative of the configurations that are to be expected for implementation of the Training Simulation Computer Complex. The following subsections present the results of the software load analysis. Section 5.2.1 presents the results of the analysis for the single CPU configuration implementation of the Simulation Software. The dual CPU analysis is presented in Section 5.2.2, while Section 5.2.3 presents the results of the four CPU configuration analysis.

The dual and four CPU configurations, although labeled as they are, can also be interpreted to represent the software loads for a configuration consisting of two and four separate dedicated computers. The software loads are presented in terms of the Reference Computer (Reference 1) for two mission phases, Ascent and On-Orbit, assuming the conventional approach sizing data. Although the On-Orbit mission phase represents the worst case estimate for the Simulation Software, its load provides several attractive options for allocating major module software packages into separate dedicated computers in addition to providing several options to reduce the fidelity of the software modules where the impact on the training requirements is not considered critical. Selection of these options reduces the Simulation Software load for the On-Orbit mission phase, thus shifting the worst case estimate to the Ascent mission phase.

The data presented in these analyses is not to be interpreted as the final memory or central processor requirements for the host computer. Additional factors such as operating system overhead, and memory allocation for the system software will be discussed and included in the definition of the hardware requirements in Section 5.5. The analysis presented here is intended to demonstrate the extent that load leveling of the Simulation Software is feasible, and to identify the Simulation Software contribution to the hardware requirements.

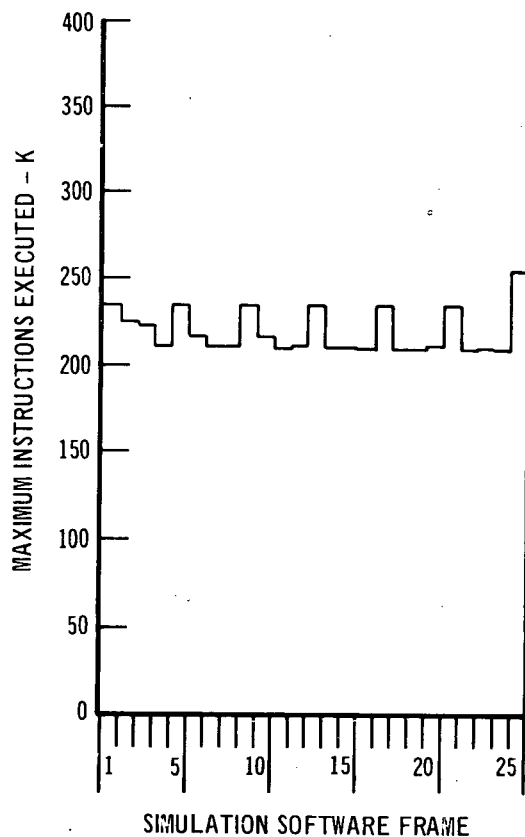
5.2.1 Real Time Software Load - Single CPU Configuration

The software load for the single CPU configuration was developed by distributing each of the 114 simulation software modules among the 25 real time execution frames. A digital computer program, LOADEM, was developed to support this large data manipulation problem. The distribution criteria used by LOADEM attempts to provide an even distribution of maximum instructions executed in each frame while obeying the specified execution rate (25, 12, 6, 3 and 1) of the module. The program assumes that there is no interdependency between modules for execution. A functional description of program LOADEM may be found in Reference 20.

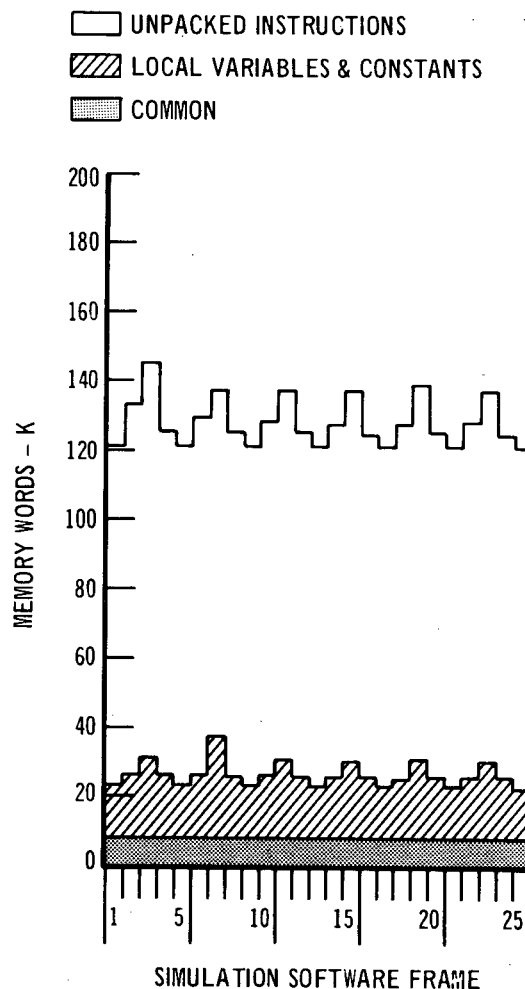
The results of the software load analysis for the single CPU configuration are presented in graphical form in Figure 6-3 and Figure 6-4. Included in these figures are a frame by frame description of the instructions executed, memory utilization, and input/output data transfer for the Ascent and On-Orbit mission phases, respectively.

The total memory requirement as presented in the figures assumes that each of the 114 Software Modules is resident in executable memory at all times. For several of the candidate computers of interest, this requirement may be too severe; therefore, vendors may elect to recommend an overlay system to satisfy the SMS processing requirement.

SIMULATION SOFTWARE LOAD DISTRIBUTION



SIMULATION SOFTWARE MEMORY UTILIZATION



SIMULATION SOFTWARE INPUT/OUTPUT WORDS

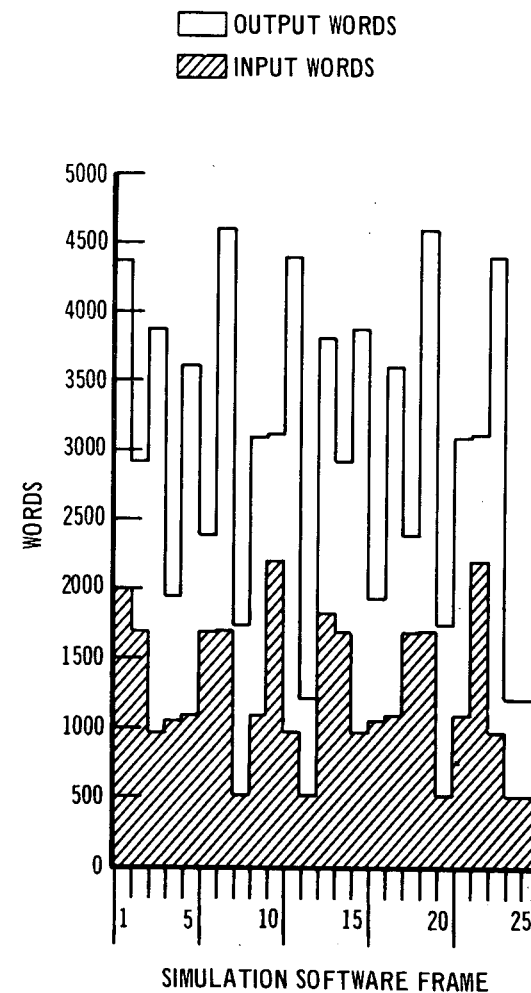


Figure 5-3 Simulation Software Distribution Summary
Ascent Mission Phase - Single CPU Configuration

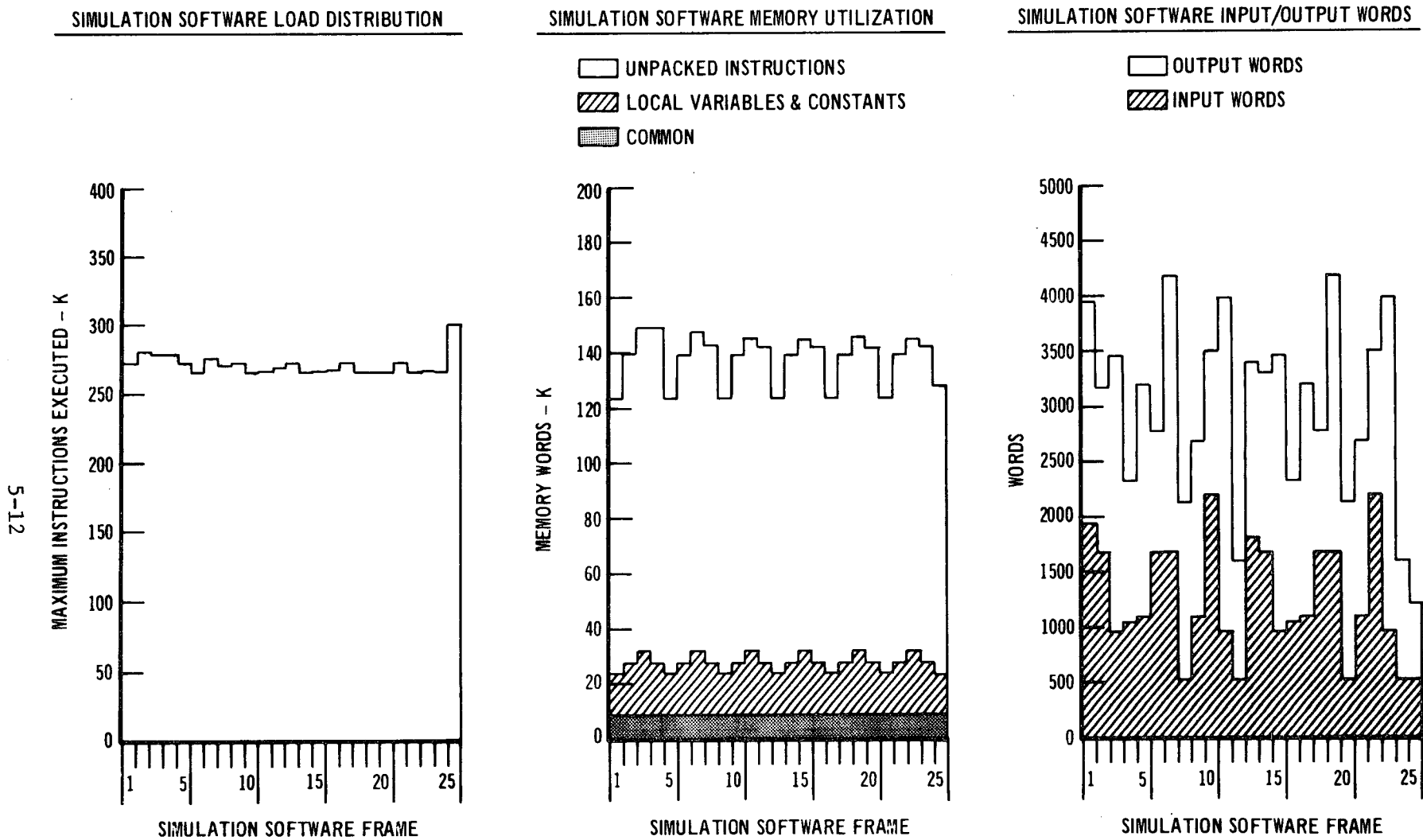


Figure 5-4. Simulation Software Distribution Summary
On-Orbit Mission Phase - Single CPU Configuration

DEVELOPMENT OF SIMULATION COMPUTER COMPLEX SPECIFICATION

A tabular representation of this data may be found in Reference 20 which includes tabulations by frame, by computation frequency and by software module for one, two and four CPU configurations.

5.2.2 Real Time Software Load - Dual CPU Configuration

The software load distribution for the dual CPU configuration has been developed. Major module categories have been grouped into each CPU along with dependent or associated module categories while attempting to maintain an even MIPS loading between the two CPU's. The simulation software modules contained in these two divisions were distributed among the 25 real time executive frames.

The results of the Software load analysis for the dual CPU configuration are presented in graphical form in Figures 6-5 through 6-8. Included in these figures is a frame by frame description of the instructions executed, the memory utilization, and the input/output data transfer for implementing both CPU's for the Ascent and On-Orbit mission phases, respectively.

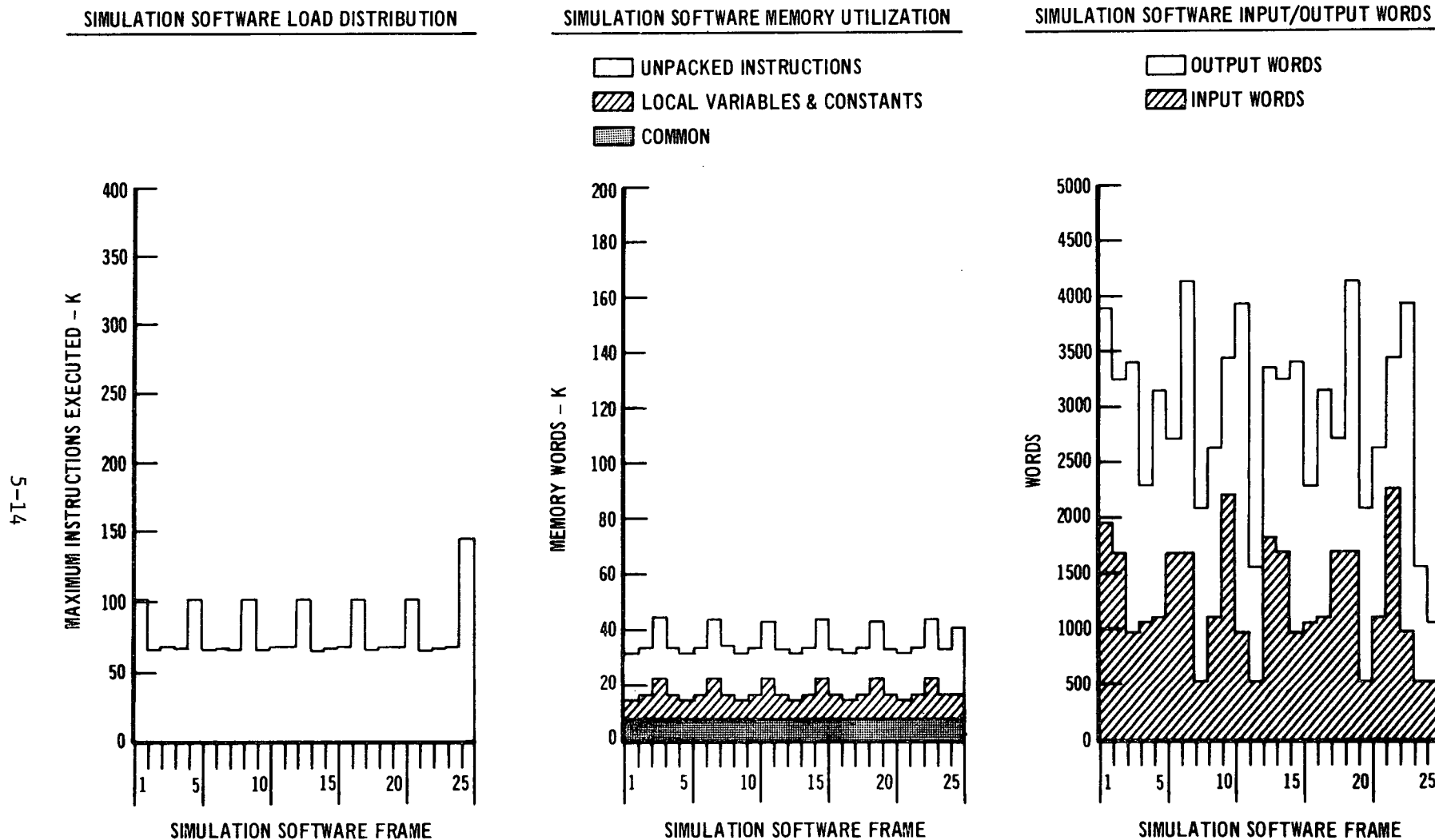
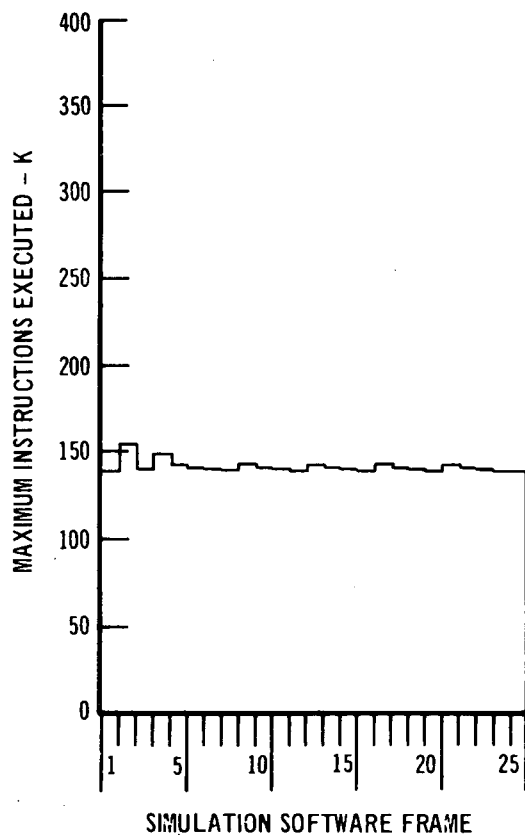
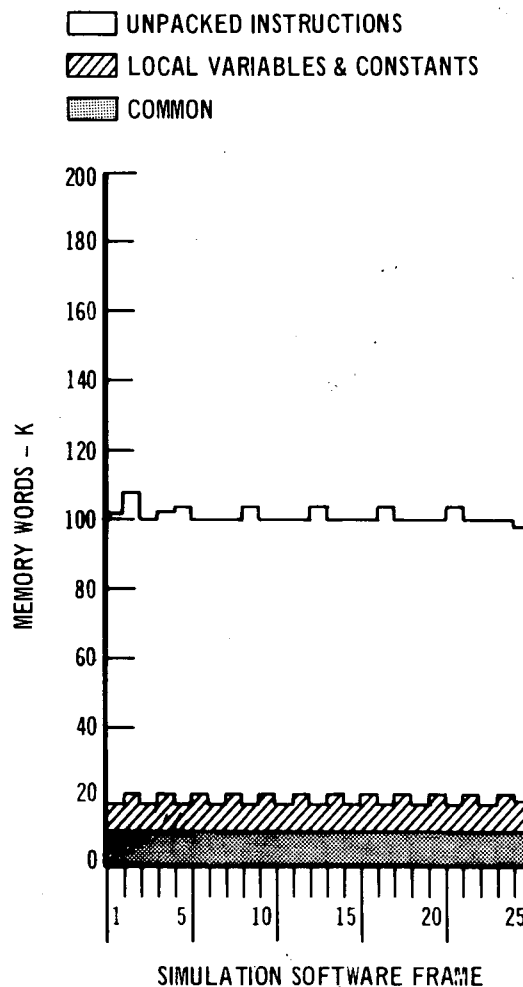


Figure 5-5, Simulation Software Distribution Summary
Ascent Mission Phase - Dual CPU Configuration - CPU No. 1

SIMULATION SOFTWARE LOAD DISTRIBUTION



SIMULATION SOFTWARE MEMORY UTILIZATION



SIMULATION SOFTWARE INPUT/OUTPUT WORDS

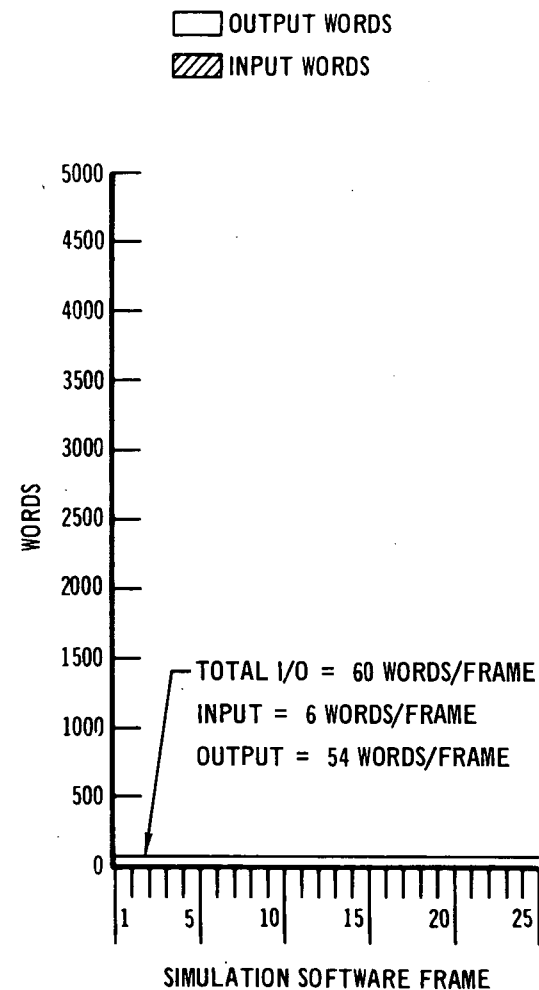
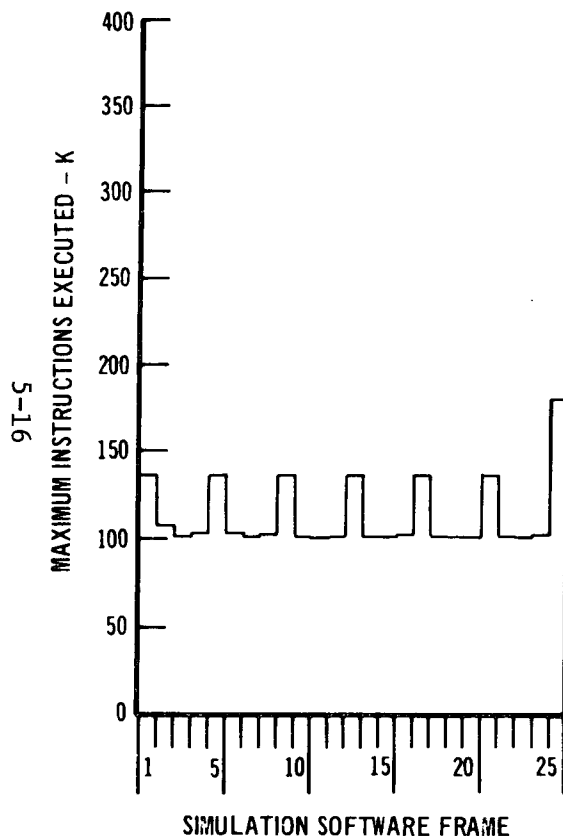
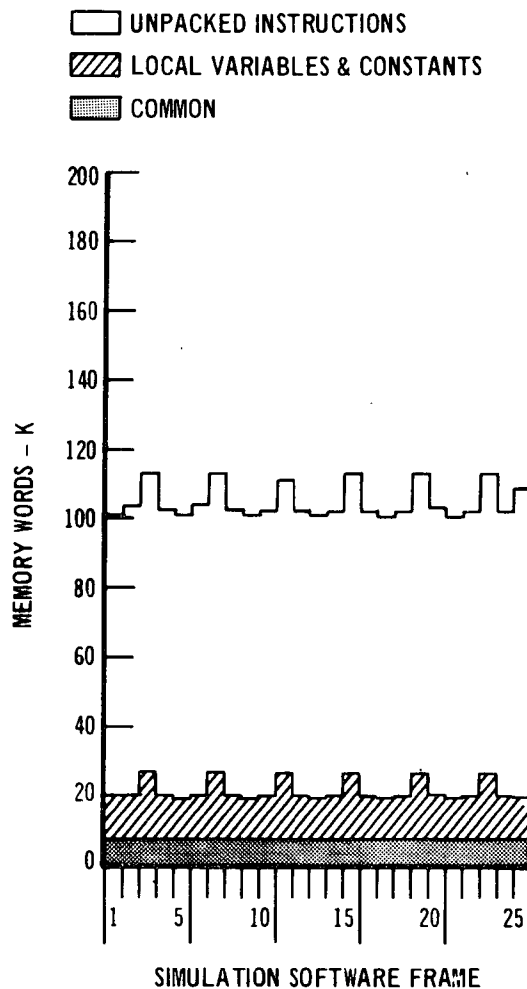


Figure 5-6. Simulation Software Distribution Summary
Ascent Mission Phase - Dual CPU Configuration - CPU No. 2

SIMULATION SOFTWARE LOAD DISTRIBUTION



SIMULATION SOFTWARE MEMORY UTILIZATION



SIMULATION SOFTWARE INPUT/OUTPUT WORDS

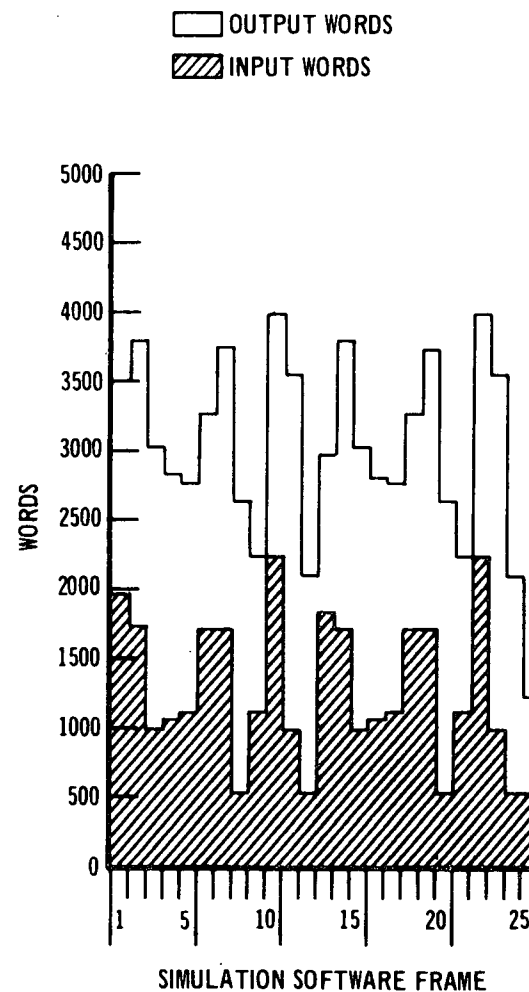
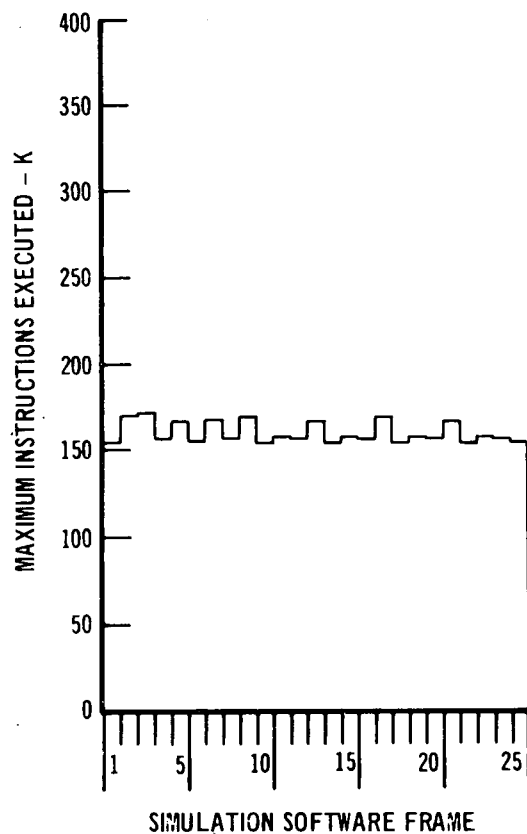
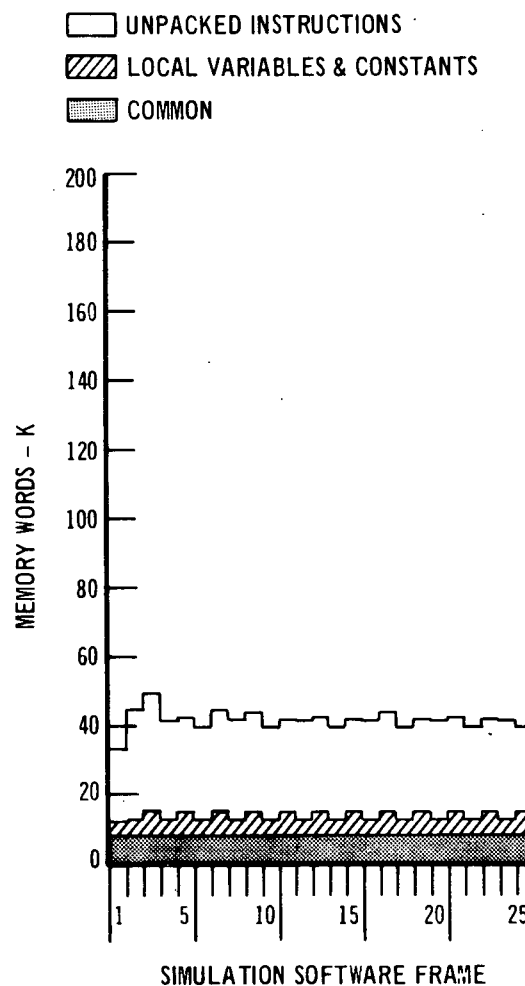


Figure 5-7 : Simulation Software Distribution Summary
On-Orbit Mission Phase - Dual CPU Configuration - CPU No. 1

SIMULATION SOFTWARE LOAD DISTRIBUTION



SIMULATION SOFTWARE MEMORY UTILIZATION



SIMULATION SOFTWARE INPUT/OUTPUT WORDS

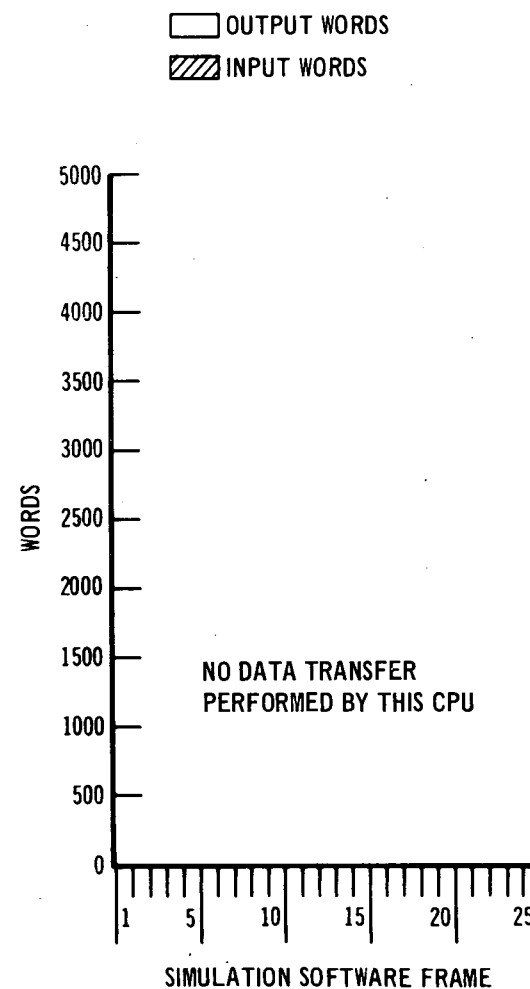


Figure 5-8 Simulation Software Distribution Summary:
On-Orbit Mission Phase - Dual CPU Configuration - CPU No. 2

5.2.3 Real Time Software Load - Four CPU Configuration

The software load distribution for the four CPU configuration has also been developed. As in the dual CPU configuration, the development of this division placed dependent module categories into the same CPU, while attempting to maintain an even MIPS loading between the four CPU's.

The results of the software load analysis for the four CPU configuration are presented in graphical form in Figure 5-9 through 5-16. Included in these figures is a frame by frame description of the instructions executed, the memory utilization, and the input/output data transfer for implementing the four CPU configuration for both the Ascent and On-Orbit mission phases.

A more detailed presentation of this data may be found in Reference 20. Data presented therein includes tabulations by frame, by computation frequency, by software module for the one, two and four CPU configurations.

SIMULATION SOFTWARE LOAD DISTRIBUTION

SIMULATION SOFTWARE MEMORY UTILIZATION

SIMULATION SOFTWARE INPUT/OUTPUT WORDS

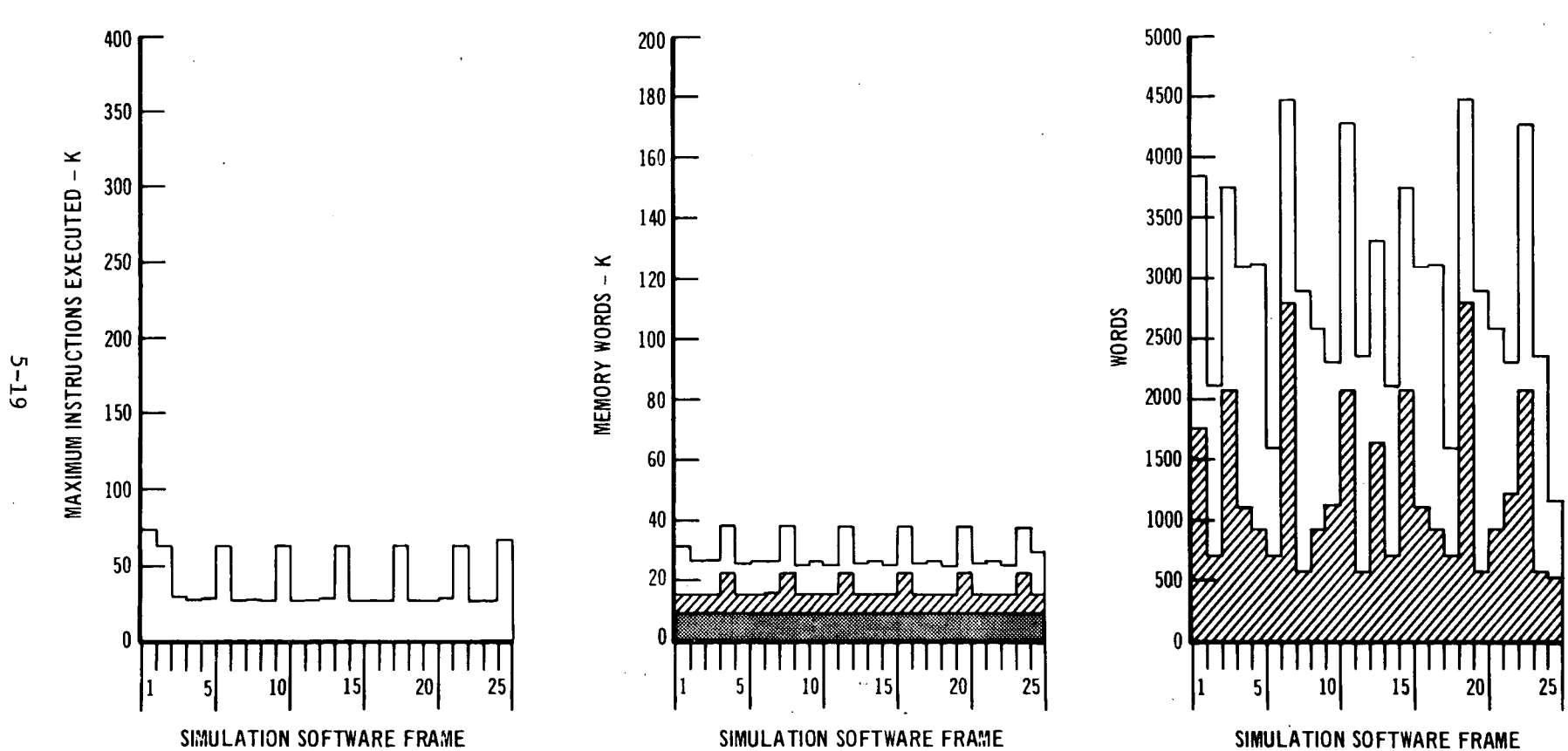


Figure 5-9 Simulation Software Distribution Summary:
Ascent Mission Phase - Four CPU Configuration - CPU No. 1

5-20

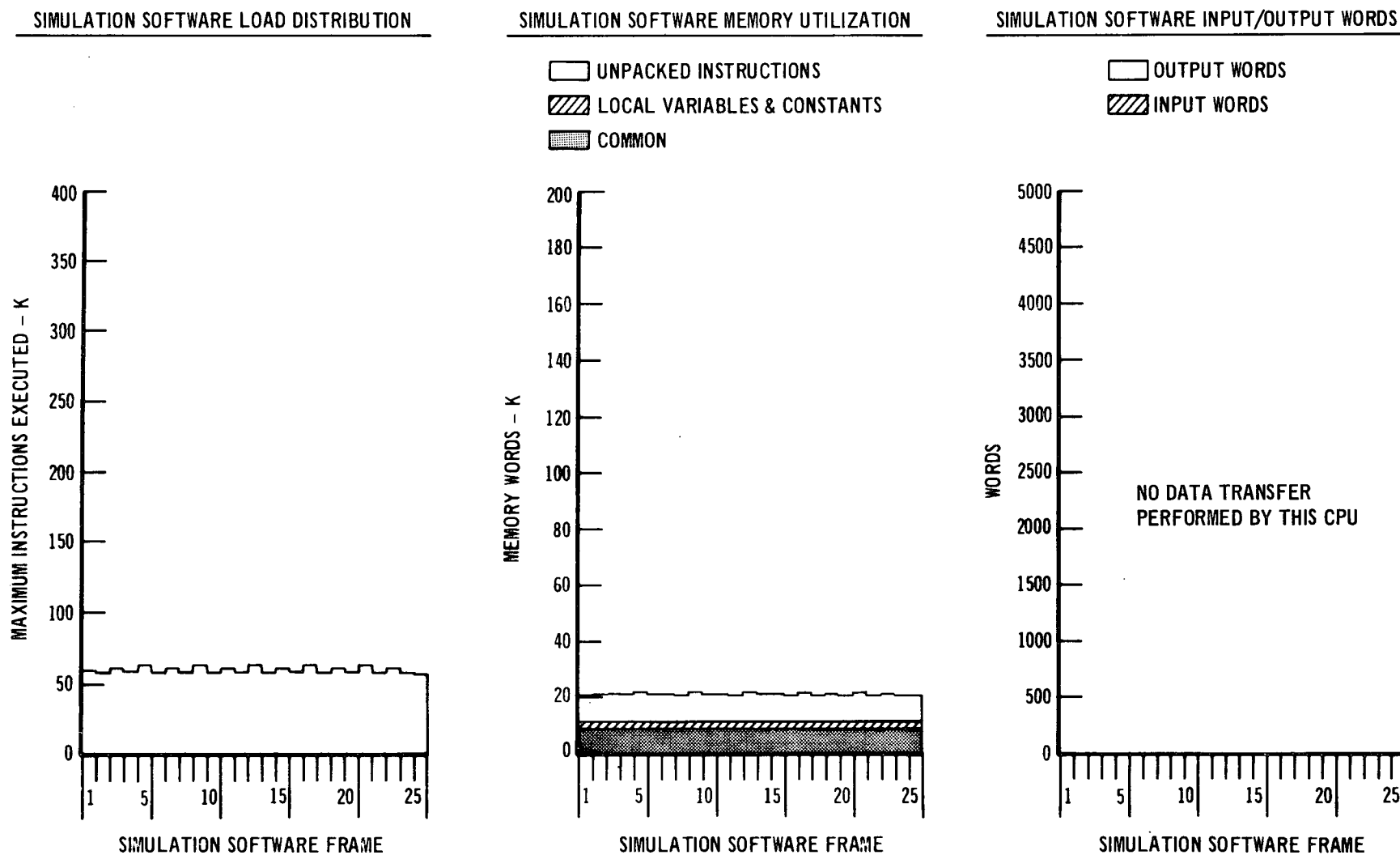


Figure 5-10 Simulation Software Distribution Summary
Ascent Mission Phase - Four CPU Configuration - CPU No. 2

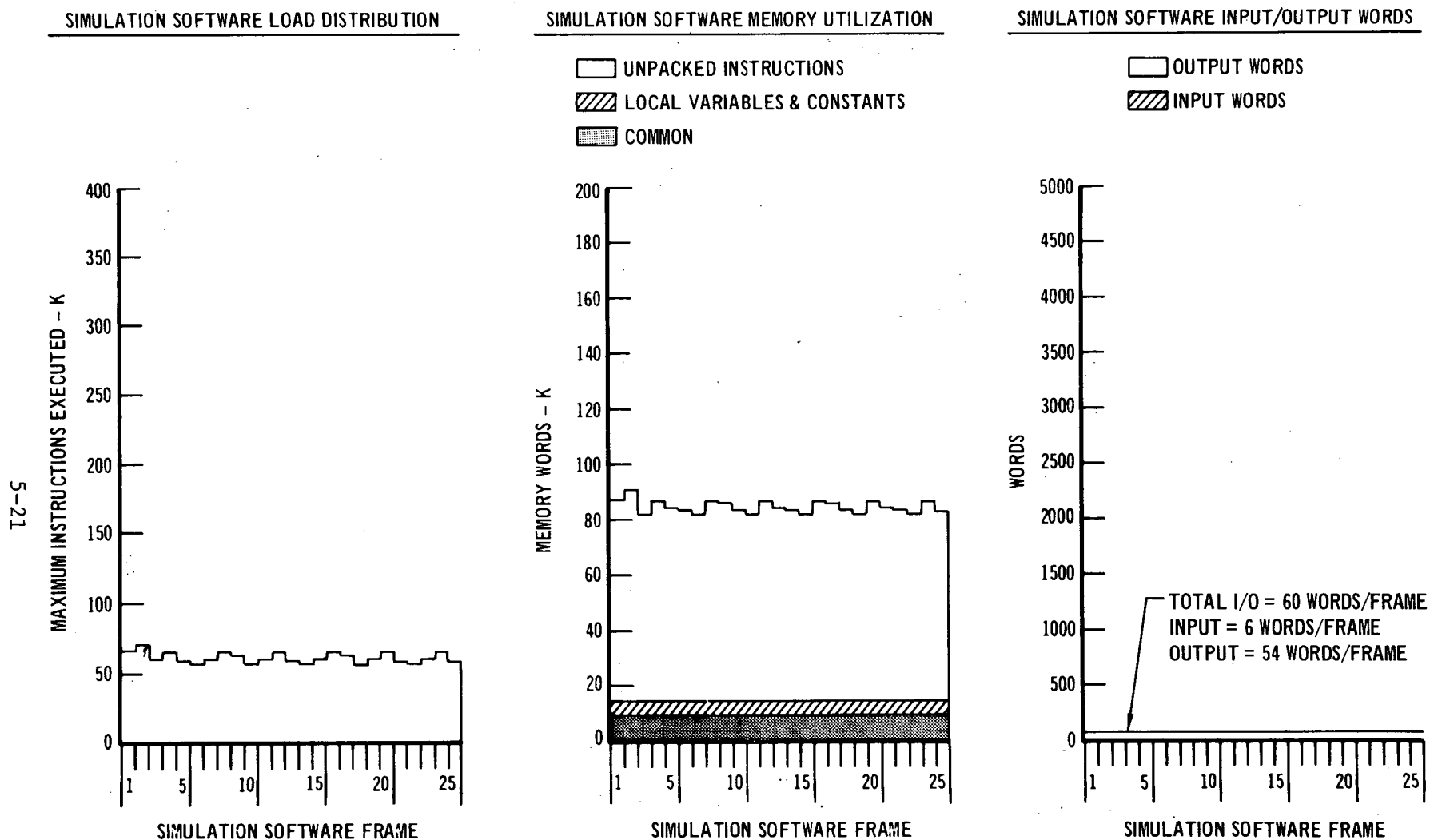


Figure 5-II. Simulation Software Distribution Summary
Ascent Mission Phase - Four CPU Configuration - CPU No. 3

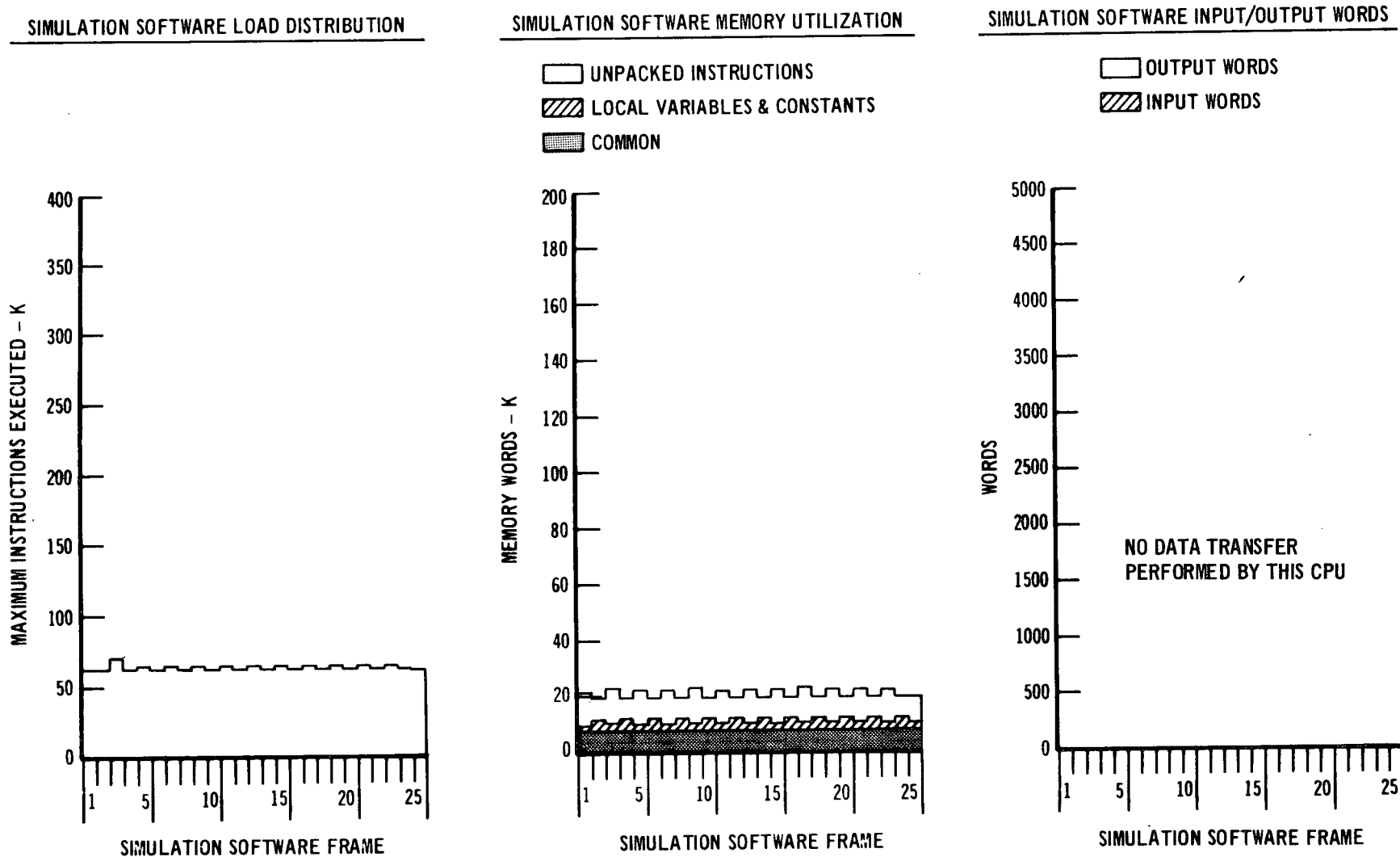
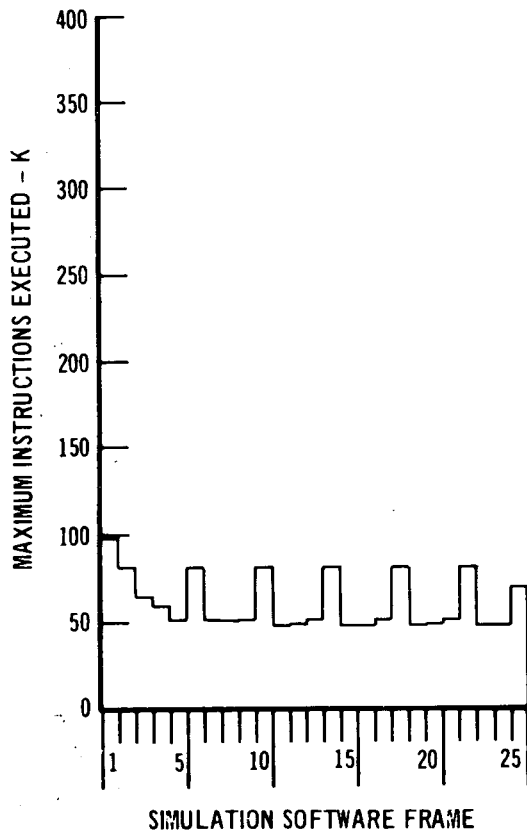
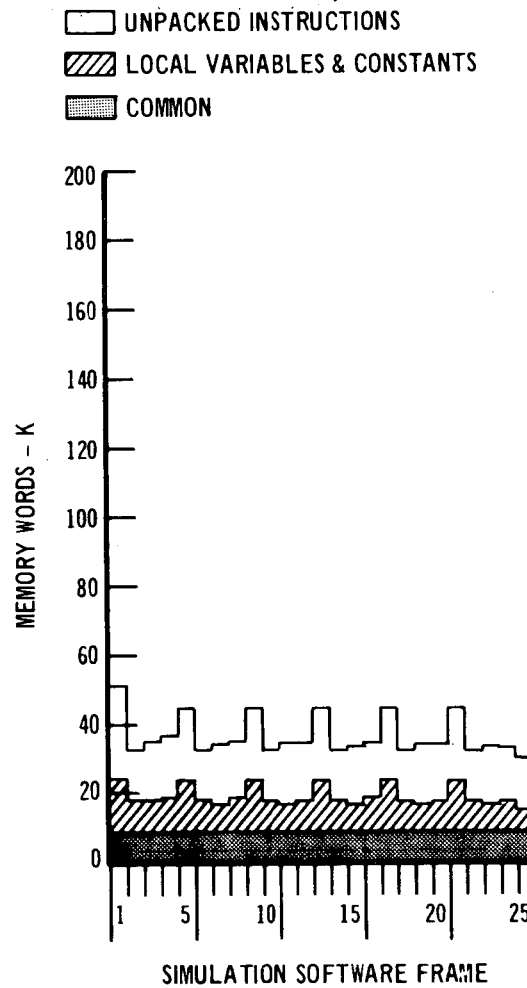


Figure 5-12 Simulation Software Distribution Summary:
Ascent Mission Phase - Four CPU Configuration - CPU No. 4

SIMULATION SOFTWARE LOAD DISTRIBUTION



SIMULATION SOFTWARE MEMORY UTILIZATION



SIMULATION SOFTWARE INPUT/OUTPUT WORDS

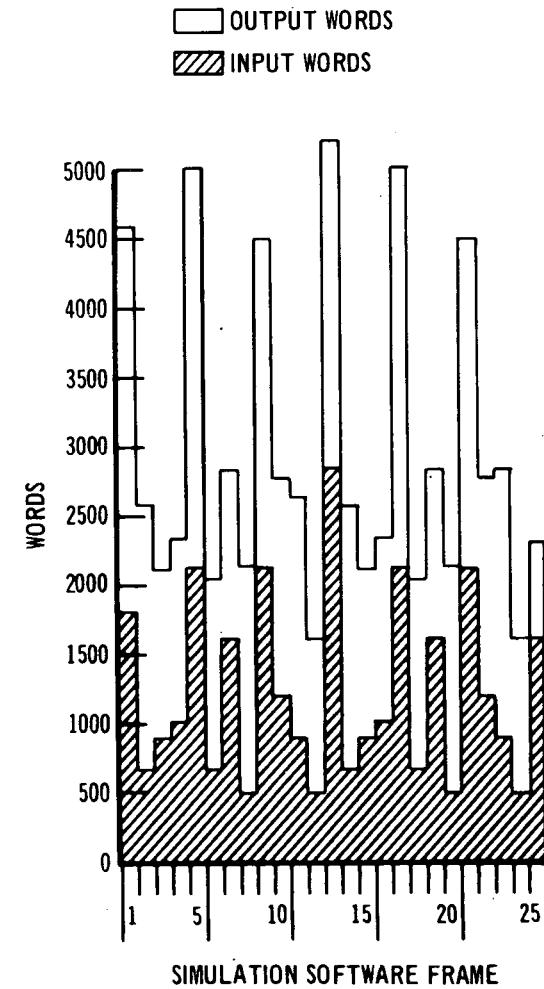


Figure 5-13 Simulation Software Distribution Summary:
On-Orbit Mission Phase - Four CPU Configuration - CPU No. 1

SIMULATION SOFTWARE LOAD DISTRIBUTION

SIMULATION SOFTWARE MEMORY UTILIZATION

SIMULATION SOFTWARE INPUT/OUTPUT WORDS

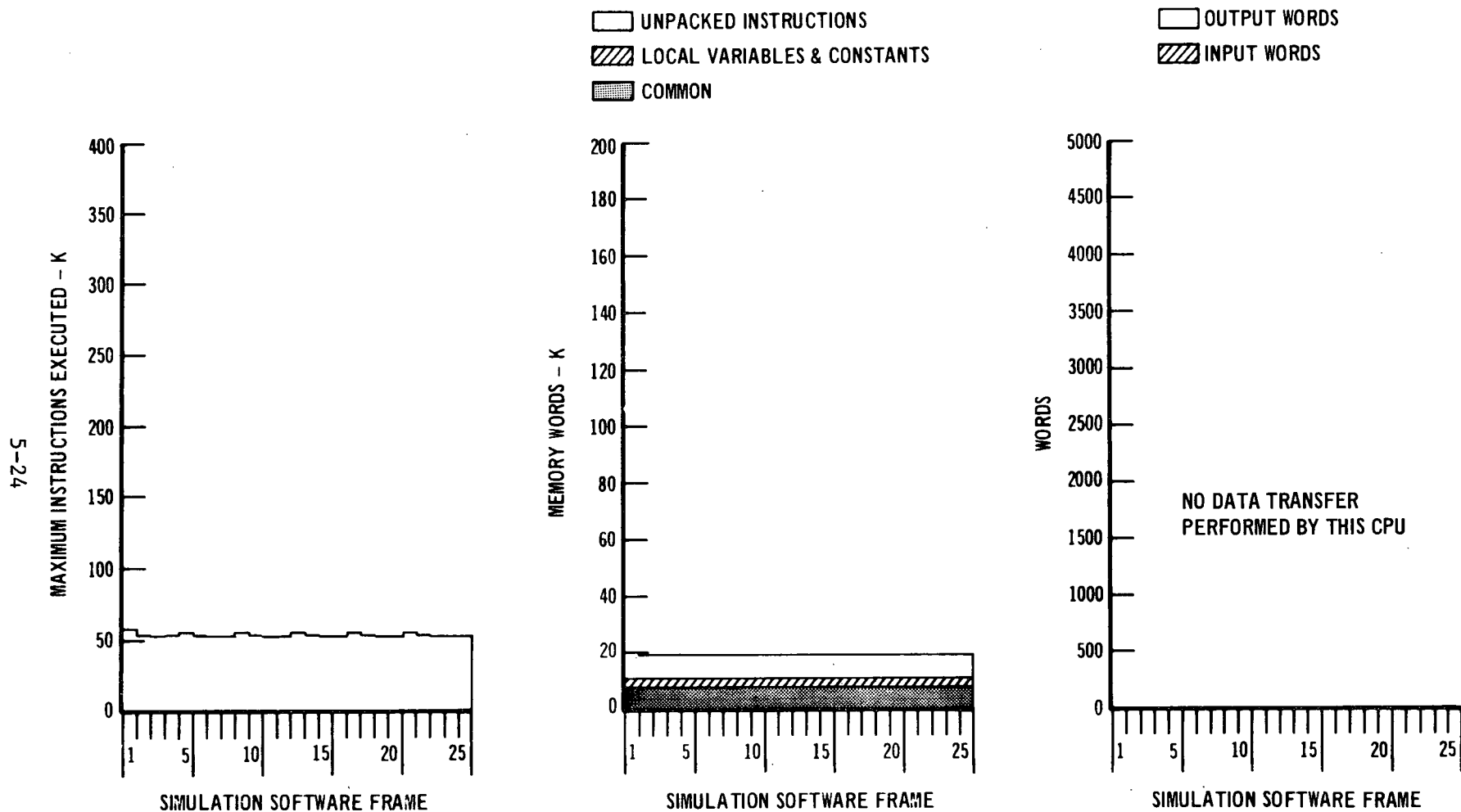
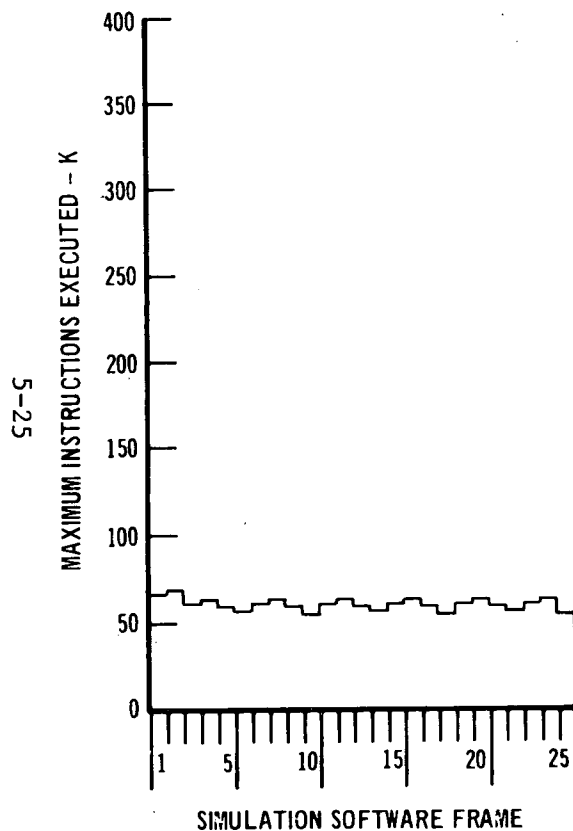


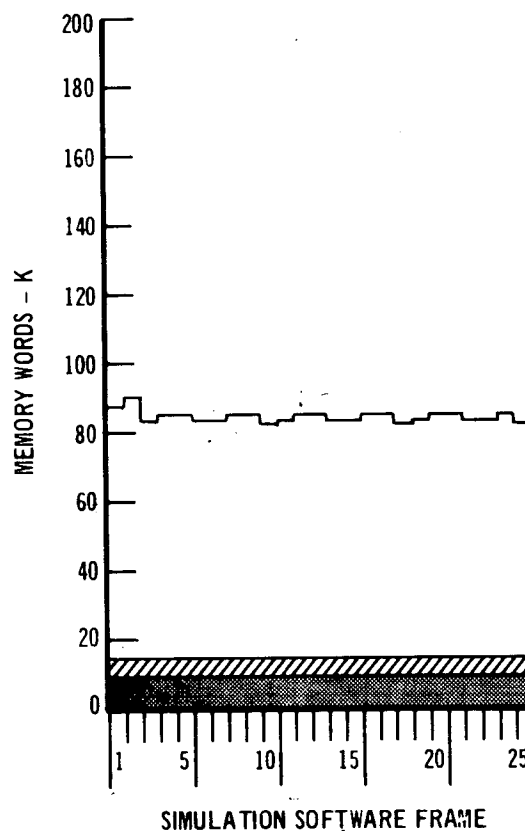
Figure 5-14 Simulation Software Distribution Summary
On-Orbit Mission Phase - Four CPU Configuration - CPU No. 2

SIMULATION SOFTWARE LOAD DISTRIBUTION



SIMULATION SOFTWARE MEMORY UTILIZATION

UNPACKED INSTRUCTIONS
LOCAL VARIABLES & CONSTANTS
COMMON



SIMULATION SOFTWARE INPUT/OUTPUT WORDS

OUTPUT WORDS
INPUT WORDS

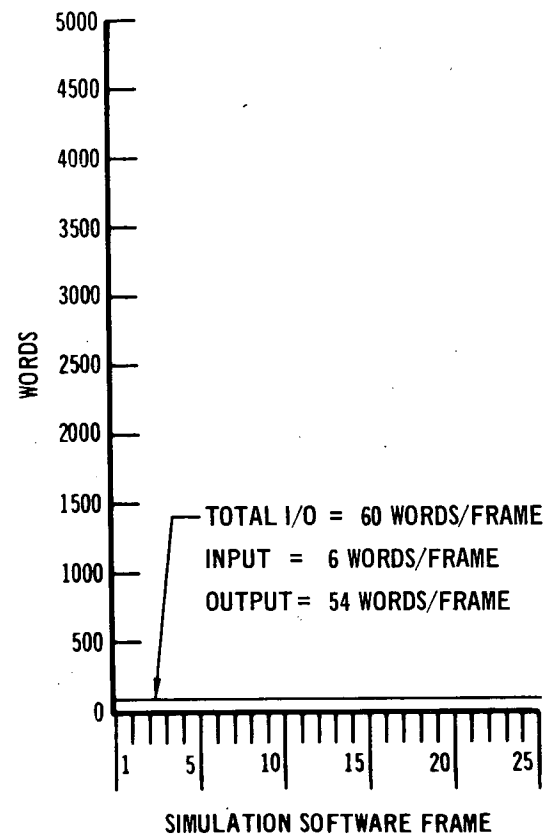


Figure 5-15. Simulation Software Distribution Summary
On-Orbit Mission Phase - Four CPU Configuration - CPU No. 3

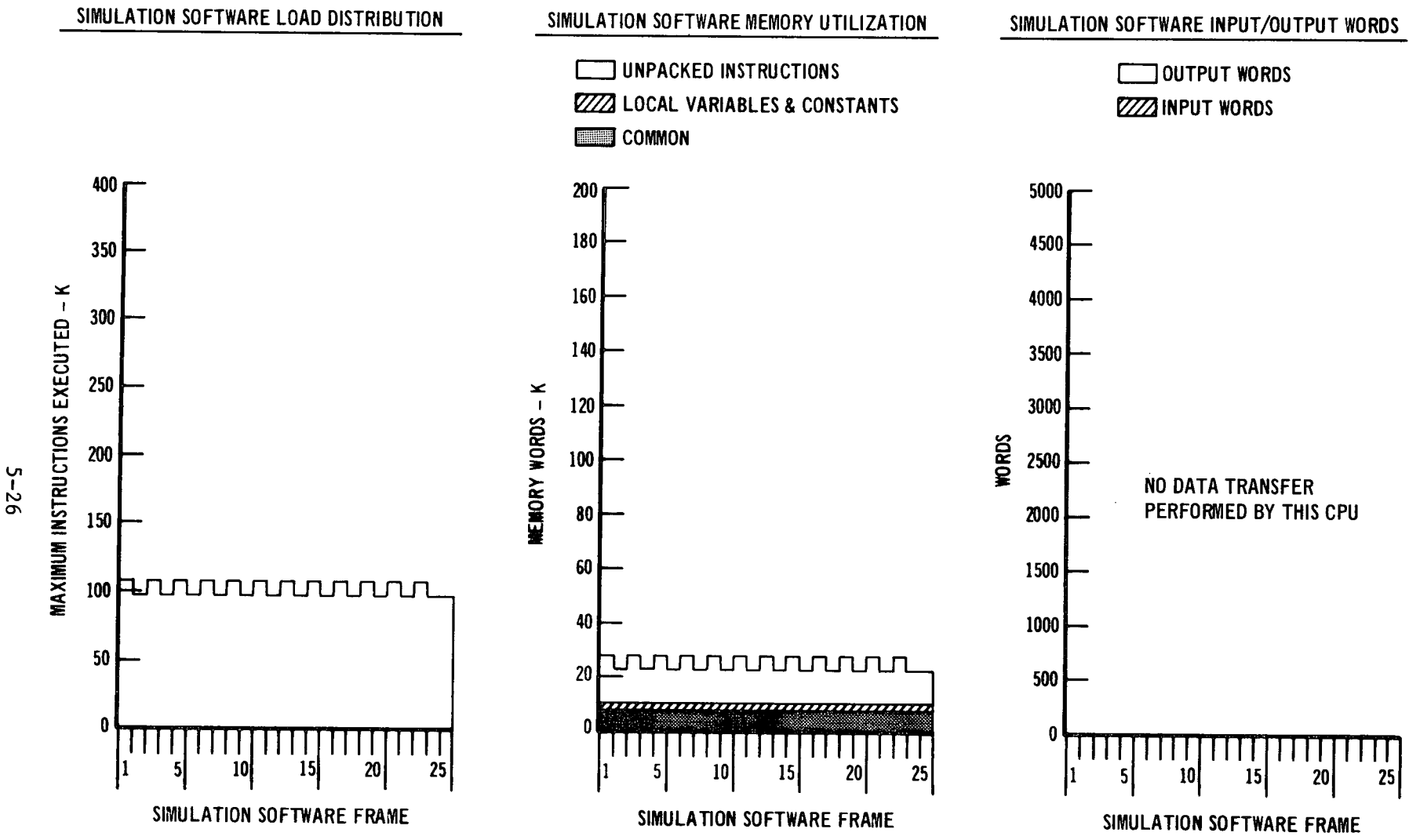


Figure 5-16 Simulation Software Distribution Summary
On-Orbit Mission Phase - Four CPU Configuration - CPU No. 4

DEVELOPMENT OF SIMULATION COMPUTER COMPLEX SPECIFICATION

5.3 REQUIREMENTS SPECIFICATION - THE SHUTTLE MISSION SIMULATOR SIMULATION OPERATIONS MIX

The simulation software loading for the Shuttle Mission Simulator was defined in Reference 1 in terms of millions of instructions to be executed per second (MIPS) and in terms of instructions and data requiring storage in memory. Instruction counts are sensitive to the hardware and system software of the particular vendor being considered. This was recognized during the simulation software sizing task and a Reference Computer was defined to which all data could be adjusted. The resulting data was quite accurate and realistic for that particular computer with its unique instruction set and associated compiler efficiencies. This Reference Computer is basically a Control Data 6000 Series computer in its instruction repertoire and word size. In defining generalized requirements for central processor capabilities, the unit of MIPS must be translated into a computer-independent quantity. The computer-independent unit which was chosen to present the CPU requirements for the Training Simulation Computer Complex is FORTRAN "operations". These operations have been defined based upon a study of existing real-time space simulation FORTRAN software. The defined operations were identified and counted in samples of existing software. The distribution of these operations was used to define a Simulation Software Operations Mix. A conversion factor was developed which will convert the MIPS requirements which were previously established to computer-independent MOPS (million operations per second) requirements.

5.3.1 Definition of Shuttle Mission Simulator Software Operations Mix

The computer-independent quantity which was selected to express the processing requirement was a FORTRAN operation. The operations which were defined were based upon the FORTRAN language since the SMS simulation software will be written in that language. The operations were identified by studying existing real-time simulation software written in FORTRAN. Five software modules were used in the analysis. Four of them are modules which are part of the CMPS simulation software and one is an existing MDAC simulation benchmark program. Table 5-2 defines these programs.

The FORTRAN software contained in these programs was studied in order to identify operations which were truly representative of real-time space simulation coding. When specifying the operations, an attempt was made to identify operations which could be easily interpreted and recoded to form a generalized benchmark FORTRAN program.

Table 5-2
Software Used to Define FORTRAN Operations and
Simulation Software Operations Mix

MODULE NAME	DESCRIPTION
SCSTVC	CMPS module which simulates the Thrust Vector Control portion of the Stabilization and Control System of the Apollo Command Module. Simulation of a hardware subsystem.
SCSACS	CMPS module which simulates the Attitude Control System portion of the Stabilization and Control System of the Apollo Command Module. This module provides a good simulation of a hardware subsystem.
DAP	CMPS module which simulates the autopilot contained within the Apollo flight software. Although this is a flight software module, it is representative of the type of software which simulates hardware subsystems.
EMS	CMPS module which simulates the Entry Monitor System of the Apollo Command Module. A hardware subsystem simulation.
ACE	A simulation benchmark program providing a six degree of freedom simulation of a Gemini vehicle during launch, ascent and orbit insertion.

After identifying and defining the FORTRAN Operations, the mix of these operations anticipated in the overall SMS Simulation Software load was determined. A hand count of the five modules was made to identify the number of each of the defined operations in this sample software. A total of 3814 operations were counted in the five modules.

Table 5-3 presents the mix in terms of percent of total for the operations defined. Several clarifications should be made about the mix. For the Subroutine Calls, the average number of arguments observed in the sample was three per call. In the tabulation of the Matrix and Vector Functions the dimensions observed were all three by three and three by one.

In preparing the mix, reference was made to work previously reported concerning software mixes. Probably the best known of the existing software mixes is the Gibson III Mix, which is most commonly called the Gibson Mix. This mix was prepared by weighting two older mixes, one prepared from data collected in March, 1959 and the second prepared from data collected in June, 1955. The second and older mix had no floating point operations. Both original mixes and the resulting Gibson III Mix are instruction mixes rather than operations mixes. Reference 21 has, however, presented an interpretation of the Gibson III Mix in terms of operations. Although the Gibson III Mix is obviously not applicable to a real-time space vehicle simulation problem in several areas, it remains a good reference point in many respects. In order to determine the reasonableness of the SMS Simulation Software Operations Mix, it was compared to the operations version of the Gibson III Mix. Table 5-4 presents this comparison.

The differences between the two mixes are obvious, but in all cases explainable. In the arithmetic operations, the SMS mix has all its arithmetic operations concentrated in floating point operations with practically no integer operations. Considering the scientific nature of the SMS software, this is understandable. The total SMS arithmetic operations (20.9%) are less than the total Gibson arithmetic operations (28.7%) for two reasons. The business nature of the software in the Gibson mix most likely caused the large amount of integer operations. These operations do not shift to floating point operations when the problem shifts to a scientific one, but rather they are eliminated. Secondly, more SMS arithmetic operations do exist but are included in other categories (DO Loop replacements, DO Loops, Standard Functions, Matrix and Vector Functions, and Inline Functions).

Indexing operations of the two mixes vary widely. The business related software of the Gibson III mix undoubtedly accounts for a large portion of those operations. The SMS, with a scientific computing load, has less need for indexing and indirect addressing. Due to this difference in the nature of the software sampled, the wide

Table 5-3
Shuttle Mission Simulator Simulation Software Operations Mix

OPERATION	PERCENT OF TOTAL	
Floating Point		20.7
+	5.1	
-	5.4	
x	8.5	
/	1.7	
Integer		0.2
+, -, *, /	0.2	
Indexing		8.6
Compares		4.1
Testing		12.4
Arithmetic	2.0	
Logical	10.4	
Boolean		11.3
AND	4.7	
OR	3.0	
NOT	3.6	
DO Loops (general)		0.5
Jumps		5.3
GO TO	5.0	
Computed, Assigned GO TO	0.3	
Subroutine Calls		0.9
Standard Functions		1.5
SQRT, SIN, COS, ATAN2	1.3	
OTHER	0.2	
Matrix and Vector Functions		0.6
Cross Product, $M \times V$, $V \pm V$	0.4	
Other	0.2	
Inline Functions		2.0
ABS	0.6	
SIGN	0.3	
Other	1.1	
Storage Moves		0.6
Replacement		31.3
Floating Point	24.2	
Integer	1.4	
Logical	5.7	
		<hr/>
		100.0

Table 5-4
Comparison of Gibson III Mix and Shuttle Mission Simulator Operations Mix

GIBSON III MIX		SMS SIMULATION SOFTWARE OPERATIONS MIX	
OPERATION	PERCENT OF TOTAL	PERCENT OF TOTAL	OPERATION
Floating Point Arithmetic +,-,*,/ All in memory A op B=C	13.1	20.7	Floating Point +,-,*,/
Integer Arithmetic +,-,*,/ All in memory J op K=L	15.6	0.2	Integer +,-,*,/
Address Modification Indexing and Indirect Addressing	26.8	8.6	Indexing
Compare Two words or digits and set indicator	4.9	4.1	Compares
Conditional Branch	9.2	12.4	Testing
Shift, and logical AND or OR	4.5	11.3	Boolean
Unconditional Branch	8.1	5.3	Jumps
Memory References More one word of fixed or floating point to accumulator and store accumulator	15.0	31.3	Replacement
Storage Moves 500 contiguous memory words to another 500 contiguous words	1.4	0.6	Storage Moves
500 random memory words to 500 contiguous words	1.4		
	-	0.5	DO Loops (general)
	-	0.9	Subroutine Calls
	-	1.5	Standard Functions
	-	0.6	Matrix and Vector Functions
	-	2.0	Inline Functions
	100.0	100.0	

5-31

difference in the mix is not unreasonable.

The two mixes agreed well in the areas of compares, testing, and jump operations. The SMS mix was noticeably higher in Boolean operations. The large amount of circuit breaker, switch position and general configuration testing in the hardware subsystem models contributes to this Boolean count in the SMS mix.

Memory references, or replacements, also varies between the two mixes. This is due to the fact that the SMS mix tabulated all memory storages in this category, whereas the Gibson III mix included a good portion of the memory stores in the floating point and integer arithmetic operations.

Storage moves, although contributing small amounts to both mixes, do vary by a large amount. The difference again is most likely due to the business related operations of the Gibson III mix.

Several operations were defined for the SMS mix which had no direct counterpart in the Gibson III mix. These include general DO Loops, subroutine calls, standard function, matrix and vector functions, and inline functions. Definition of these operations contributed useful detail to the SMS mix. If these operations were not defined, these percentages would have contributed to the counts of other operations, particularly the arithmetic operations, thus bringing them even closer to the Gibson III mix data.

5.3.2 MIPS to MOPS Conversion Factor

The SMS Software Operations Mix just presented defined the details of the computing required of the TSCC in terms of computer-independent FORTRAN operations. The overall computing capacity of the TSCC must also be specified in terms of a computer-independent quantity. The operations which have been defined will be used as this new quantity when expressed in terms of MOPS, million operations per second.

The unit of MIPS which has been used in all sizing estimates and software load analyses required conversion to the new MOPS unit. Since re-estimation of the software load is impractical and, in reality, not required, development of a suitable conversion factor allowed previously computed estimates and analyses to be used in developing the requirement for CPU processing.

DEVELOPMENT OF SIMULATION COMPUTER COMPLEX SPECIFICATION

Two approaches were followed in developing the required conversion factor to units of MOPS, an analytical approach and a software benchmark approach. The analytical approach was based upon developing the conversion factor from three ratios which were able to be determined from available data. The following expression indicates how the three ratios were combined to yield the required conversion factor:

$$\text{Conversion Factor} = \frac{\text{OPS EXEC}}{\text{INST EXEC}}$$

$$\frac{\text{OPS EXEC}}{\text{INST EXEC}} = \frac{\text{OPS STR}}{\text{INST STR}} \times \frac{\text{INST STR}}{\text{INST EXEC}} \times \frac{\text{OPS EXEC}}{\text{OPS STR}}$$

where

OPS EXEC = Operations Executed by the SMS software taking into account loops, branches, calls to ther modules, etc.

INST EXEC = Instruction Executed (Reference Computer machine instructions) by the SMS software taking into account loops, branches, calls to other modules.

OPS STR = Operations Stored in memory for the SMS software.

INST STR = Instructions Stored (Reference Computer machine instructions) in memory for the SMS software.

The final results of this analysis were obtained when these factors were combined for the worst case, On-Orbit.

$$\frac{\text{OPS EXEC}}{\text{INST EXEC}} = \frac{1}{3.85}$$

A second approach was taken to determining the same factor. The operations mix defined in Table 5-3 was used to construct a benchmark program coded in FORTRAN. This program was compiled and executed on the CDC 6400 located in Building 35 of JSC. The program executed exactly 1000 operations as defined by the mix and tabulated the time required to complete the processing. Since the CDC 6400 is identical to the defined Reference Computer, the results of this benchmark program lead to the same conversion factor. Over the span of one million operations, the program required an average of 6.012 μ seconds per operation. Assuming 1.29 μ second per average instruction executed on the CDC 6400, an observed average instruction time for simulation programs, the benchmark results in the ratio:

$$\frac{\text{OPS EXEC}}{\text{INST EXEC}} = \frac{1}{4.66}$$

The results of these two approaches to defining the required MIPS to MOPS conversion factor have lead to two numbers which, though not in perfect agreement, do agree adequately. Of the two approaches used in estimating the conversion factor, the benchmark approach is felt to be the more accurate. However, realizing that the amount of experience behind both methods of analysis is small due to the newness of the operations mix concept, a conservative approach has been followed and the recommended estimates of the MIPS/MOPS ratio is as follows:

$\frac{\text{MOPS}}{\text{MIPS}}$	=	$\frac{\text{OPS EXEC}}{\text{INST EXEC}}$	=	$\frac{1}{4}$
-----------------------------------	---	--	---	---------------

5.4 COMPUTER SIMULATION ANALYSIS

A digital computer system simulation has been used to supplement and verify engineering analysis of computer hardware requirements for the Training Simulation Computer Complex. Model development was facilitated by the use of a general computer simulation program, COMPSIM. Details of COMPSIM and the models prepared for this study are contained in Reference 20 along with a general description of the simulation and analysis of the results.

COMPSIM is a TRW computer simulation tool which combines the general system modeling capabilities of discrete simulation languages such as GPSS and SIMSCRIPT with a set of functional operators specifically designed for the modeling of software-software and software-hardware interactions.

The key to the flexibility of COMPSIM is that all functions are provided as FORTRAN subroutines so that the user has access to all standard FORTRAN capabilities when preparing his models. This feature also simplifies the process of modifying or expanding COMPSIM.

These capabilities were used to implement five categories of models for the computer simulation activities. The following describes these capabilities:

Data Flow Model (SMPIO). This model simulates all time delays associated with peripheral I/O operations, generates device end interrupts to the CPU model, and reactivates tasks that have been delayed waiting for I/O.

Interrupt Driver Models. Two drivers generate interrupts when real-time tasks should start and when a new batch is ready to be copied to the job file.

General Operating System Models. Those functions basic and common to all operating systems (task scheduling, interrupt processing, resource allocation, etc.) have been identified and appropriate models are included in the simulation. Descriptions of the overall operating system and individual models are contained in Reference 20.

Real-Time Program Models. These models represent the software load required to support the Shuttle Mission Simulator.

Batch Program Models. These models represent the background non-real-time software load and determine CPU utilization in the batch mode. The major routines in this category are compiler, loader, editor, and the general batch model, all of which are described in Reference 20.

These models cause continuously changing competition for CPU and I/O hardware resources. New active tasks are generated each time a clock interrupt is received or a batch job is processed by the batch monitor and resource allocator. Each active task operates in a unique manner as specified by its program model. When a task reaches highest priority, the scheduler gives it control of the CPU. After some processing time, the task will request I/O and, therefore, become inactive until the operation is completed. The next highest priority task will then get the CPU. When the I/O operation completes, the delayed task becomes an active task once more. If it has a higher priority, it will preempt the currently executing task.

Even though simulations of the COMPSIM type are more attuned to a detailed analysis of a specific system or comparative evaluations of alternate configurations, valuable information can be obtained for requirements definition. Even with a limited run scheduled, certain specific objectives can be accomplished. Since the number and types of I/O operations and the number of instructions required to perform the real-time tasks are well defined, the minimum CPU capability can be established for alternate channel configurations. The representative batch load can then be added to the real-time tasks to identify resource conflicts, data traffic bottlenecks, and determine parameters that have the greatest effect on throughput.

The results of the simulation activity are discussed in Reference 20 and have been incorporated into the hardware requirements presented in the following section.

5.5 STATEMENT OF COMPUTER HARDWARE REQUIREMENTS

The computer hardware required to support the SMS will be summarized in this section. Central processor unit, memory, and input/output channel requirements are specified for single CPU configurations as well as for multiple CPU/computer configurations. The requirements for standard computer peripheral equipment for the local facility as well as for remote stations are also be defined. Additional requirements affecting the computer equipment are specified. These include requirements for reliability, maintainability, equipment expansion, acceptance testing, operating environment, computer interface and communication, and performance measuring equipment.

The requirements have been developed from the software load analysis data, the Simulation Software operations mix data, and the computer simulation data referenced in the previous sections. The results of these analyses have been interpreted and combined, and weighted with information describing typical computer equipment available today, as presented in Reference 22. The following paragraphs present this rationale and the resulting computer hardware requirements.

5.5.1 Central Processing Unit

The central processing unit of TSCC is required to execute the Simulation Software load as defined in Section 4.1. This software load consists of the real-time simulation software processing and its associated input/output, and non-real-time batch/interactive processing for simulation support jobs.

The basic computing requirements for the TSCC were spelled out in the Simulation Software Sizing Report, Reference 1. Realizing that vendors of various computing equipment may elect to propose various computer configurations to meet the software load requirements of the SMS, the analyses of the single, dual and four CPU configurations reported in Sections 5.2.1, 5.2.2, and 5.2.3 were performed. These analyses revealed that, in general, the processing load was able to be distributed evenly from frame to frame. In some cases, the processing in a particular frame exceeded any other frame. The loading on this peak frame becomes the driving item in specifying the processing capability of the CPU. It is not unreasonable to assume that the final design of the SMS may in fact contain these peak loading situations. The overall CPU requirement for these distributions is then the equivalent load produced by these peak frames.

DEVELOPMENT OF SIMULATION COMPUTER COMPLEX SPECIFICATION

For the single CPU configuration, after the software load was analyzed and distributed, the peak frame did not vary from the average loading for the twenty-five frames by a large amount as indicated in Figures 5-3 and 5-4. In the analysis of the multiple CPU configurations, the software load was further distributed among the specified number of CPU's. For the two mission phases, the average loading in the CPU's was not the same. This was due to desirability of keeping similar software processes grouped in one CPU. However, the equivalent peak loading of both CPU's for the Ascent and On-Orbit mission phases was nearly equal. Again due to the desirability of keeping related software processes in one CPU, the average load per CPU also varied over the four CPU's from the 4 CPU configurations. However, the variation for the Ascent software was much less than that of the On-Orbit mission phase. The software which simulates the Payload Manipulator System of the Shuttle vehicle is of a large enough magnitude that, even though it alone is assigned to a single CPU, it causes the average loading for the On-Orbit mission phase to be unequal. When considering the equivalent peak loading, the four CPU's for the Ascent phase are loaded almost equally, while the CPU's for the On-Orbit phase are again unequally loaded.

The computing loads just discussed have all be presented in units of MIPS, the Reference Computer loading unit. In order to provide processing requirements which are computer-independent, the conversion factor developed in Section 5.3 was applied to the data to generate processing requirements stated in terms of the previously defined MOPS. Table 5-5 summarizes the average load and equivalent peak load resulting for each of the configurations and for the Ascent and On-Orbit mission phases in terms of both MIPS and MOPS.

In addition to the discussion above concerning equivalent peak loading, a second aspect of the software load affects the processing requirements. The analysis discussed in Section 5.4 concerning the computer simulation activity has revealed that the time required to perform simulation input/output each frame is not insignificant. The peak time required for input/output varies depending on the method of communication with the five flight computers and flight CRT's. If data is passed to the flight computers in a parallel mode, the peak time required to complete all simulation I/O is 4.4 milliseconds. If data is passed to the flight computers in a serial mode, the peak time required to complete

Table 5-5,
Central Processing Unit Simulation Software Loading

(Operating System Overhead Processing Not Included)

CONFIGURATION	ASCENT MISSION PHASE				ON-ORBIT MISSION PHASE			
	CPU NUMBER				CPU NUMBER			
	1	2	3	4	1	2	3	4
SINGLE CPU								
AVERAGE LOAD MIPS	5.5				6.8			
(MOPS)	(1.4)				(1.7)			
EQUIVALENT PEAK LOAD	6.4				7.6			
	(1.6)				(1.9)			
WITH 10% I/O ALLOWANCE	7.1				8.4			
	(1.8)				(2.1)			
DUAL CPU								
AVERAGE LOAD	2.4	3.1			2.7	4.2		
	(0.6)	(0.8)			(0.7)	(1.1)		
EQUIVALENT LOAD	3.8	3.9			4.5	4.3		
	(1.0)	(1.0)			(1.1)	(1.1)		
WITH 10% I/O ALLOWANCE	4.2	4.3			5.0	4.8		
	(1.1)	(1.1)			(1.3)	(1.2)		
FOUR CPU								
AVERAGE LOAD	0.9	1.5	1.5	1.6	1.5	1.3	1.5	2.6
	(0.23)	(0.38)	(0.38)	(0.40)	(0.38)	(0.33)	(0.38)	(0.65)
EQUIVALENT PEAKLOAD	1.8	1.6	1.7	1.8	2.5	1.5	2.0	2.0
	(0.45)	(0.40)	(0.43)	(0.45)	(0.63)	(0.38)	(0.50)	(0.70)
WITH 10% I/O ALLOWANCE	2.0	1.8	1.9	1.9	2.7	1.6	2.2	3.1
	(0.50)	(0.45)	(0.48)	(0.48)	(0.68)	(0.40)	(0.55)	(0.78)

DEVELOPMENT OF SIMULATION COMPUTER COMPLEX SPECIFICATION

all simulation I/O is 6.3 milliseconds. If it is assumed that the input/output load can be altered slightly in the final SMS design such that these I/O delays do not exceed 4 ms or 6 ms for the two data transfer modes, then it is observed that either 10% or 15% of the processing frame will be devoted to I/O. Processing of the real-time software load must be accomplished in the remaining portion of the frame. As a result, the processing capability of the CPU must be great enough to complete the software processing in 85% to 90% of the frame time. Table 5-5 presents the processing loads adjusted to allow for input/output times of 10% of the frame time.

The computer simulation analysis of Section 5.4 also has indicated that the requirement of the CPU to support an average batch load of .332 MIPS can be met without imposing additional processing requirements. When the real-time input/output is occurring, that 10% of the frame can be devoted to processing the batch load. For the On-Orbit single CPU configuration, this amounts to .84 MIPS of computing power available for batch processing. For the multiple CPU configurations, a similar situation exists with each CPU available for batch processing during the input/output period. Therefore it is required that the CPU process the background batch and interactive jobs during that period when the real-time simulation is performing I/O. If for some reason the required .332 MIPS of batch processing cannot be accomplished in 10% of the frame time, more time per frame must be allocated. In that case the new overall CPU processing requirement must be calculated based upon the new I/O time allowance and a CPU of that capacity will be required.

The CPU requirements are then summarized as follows. The CPU must be capable of supporting the computing load for the simulation of worst case loading, the On-Orbit mission phase. Up to ten percent of the available frame will be allocated for completion of simulation input/output data transfers. Batch processing equivalent to .332 MIPS or .083 MOPS must be processed during the I/O period. For multiple CPU configurations, all CPU's within a configuration must be of the same capacity to allow flexibility of design and software reconfiguration potential. The CPU processing requirements are summarized below:

<u>CPU Requirements</u>		
<u>Configuration</u>	<u>MIPS</u>	<u>MOPS</u>
Single CPU	8.4	2.1
Dual CPU (Each)	5.0	1.25
Four CPU (Each)	3.1	0.78

DEVELOPMENT OF SIMULATION COMPUTER COMPLEX SPECIFICATION

Any processing requirements which are required to support the computers operating system and its activities are not included in the above CPU requirements. Bidders must add the processing requirements of their system software operating system to the above stated requirements and supply a CPU of the appropriate capacity. Section 8 defines the operations required of the TSCC system software.

The above requirements for dual and four CPU configurations also apply to dual and four computer configurations.

5.5.2 Memory

The memory requirements for the host computer of the Training Simulation Computer Complex are stated in terms of word size and memory size. Several different factors are considered in the statement of these requirements. These considerations include: the data of the software load analysis presented in Section 5.2, the results of the vendor survey presented in Reference 22, and data taken from several existing simulation facilities at the Johnson Space Center.

5.5.2.1 Word Size

Single and double precision minimum requirements for the fixed and floating point data items are as follows:

<u>Precision</u>	<u>Floating Point</u>	<u>Fixed Point</u>
Single	7 Bits Exponent 24 Bits Fraction	32 Bits
Double	7 Bits Exponent 48 Bits Fraction	60 Bits

5.5.2.2 Memory Size

The modular approach to the definition of the SMS software load, developed in the Simulation Software Sizing task, Reference 1, lends itself to implementation on a variety of computer configurations for the Training Simulation Computer Complex. The modular approach allows distribution of the software load into multiple dedicated computers or multiple dedicated central processor units in addition to a single large computer. The analysis of the software load discussed in Section 4.2 of this report is interpreted to apply to a single computer configuration, a multiple CPU and/or multiple computer configurations. For the multiple CPU configuration, it is assumed that one memory requirement exists which is shared by each CPU. For the

multiple computer configurations, each computer in the configuration must have the same memory characteristics. This rationale provides flexibility of design and flexibility of software reconfiguration. Secondly, this requirement provides a minimum reduction (worst case 50%) of the training requirement should one computer in the system fail. In some instances, although a failure may exist in one computer, training could continue for a different mission phase requiring a smaller software load. This rationale results in more than the minimal total memory required when multiple computers are configured.

In addition to the Simulation Software load, a memory requirement exists to support the non-real-time (Batch and Interactive Software) processing load. The Software Sizing Report, presents an estimate of 75000 words of storage required for this load. Review of several large analysis programs used to support the Command Module Procedures Simulator, indicate that approximately 25 percent of this estimated storage should be allotted to data items.

The memory requirements are stated in terms of the Reference Computer. The characteristics of the Reference Computer, described in Reference 1, do not pack more than one instruction per word. Each vendor should apply the appropriate instruction packing ratio for his proposed system and be required to include documentation to demonstrate this ratio.

The memory requirements for the configurations that are of interest for implementation of the computer complex are summarized as follows:

<u>Configuration</u>	<u>Data Items</u>	<u>Unpacked Instructions</u>	<u>Memory Words</u>
Single CPU	60,000	224,000	284,000
Dual CPU with shared memory	60,000	224,000	284,000
Four CPU with shared memory	60,000	224,000	284,000
Dual Computer	48,000	165,000	213,000
Four Computer	34,000	150,000	184,000

In addition to these memory requirements, each configuration must be incremented to include the proper memory allotment necessary to support the system software requirements specified in Reference 23. Each vendor must include documentation to substantiate the memory allotment for these requirements.

DEVELOPMENT OF SIMULATION COMPUTER COMPLEX SPECIFICATION

For the data items presented in the above requirements, 25 percent logical and 75 percent fixed and floating point words are required. Of the fixed and floating point words, 5 percent shall be double precision. These percentages are typical of existing large simulation programs.

5.5.3 I/O Channels

The host computer is required to communicate with the minicomputers in the computer complex in order to provide the necessary interface for the five flight computers, the flight hardware CRT's, the simulator hardware, Mission Control, and the instructor console. The minicomputers collect the data from the SMS hardware devices in buffers and transfer data to and from the host computer in one continuous block at the beginning and end of each time frame. In general, data flows from the host computer through the flight computers back to the host computer then to the crew stations, other flight computers, and other equipment. This data flow process provides for direct control of the crew station hardware and the flight computers from the host computer. There shall be no direct data transfer from flight computer to flight computer or flight computer to simulated flight instruments as in the Shuttle vehicle. This requirement provides design flexibility for introducing failures into the simulation thus providing the necessary training environment for the flight crews.

The following list summarizes the data paths and the minimum transfer rates required to support these devices:

<u>Device</u>	<u>Minimum Transfer Rate</u>
Flight Computers (5)	500K words per second each
Flight Hardware CRT's	500K words per second
Mission Control	90K bits per second
Instructor Console	200K words per second
Simulator Hardware	200K words per second

The computer simulation activities have demonstrated that a serial transfer of the data for the five flight computers via one channel is feasible; however, a slight penalty of approximately .4 MIPS additional computer capability is required due to a reduction in the percent of the frame time available for computing. Parallel transfer of data to the flight computers required less than 10% of the 40 ms frame in all cases.

It is recommended that vendors provide I/O channels of sufficient number and compacity such that data transfer, to the above specified devices operating at these minimum transfer rates, may be accomplished in 10% (4 ms) of the simulation frame.

5.5.4 Peripheral Equipment

The peripheral equipment requirements for the Training Simulation Computer Complex are stated in two categories dependent on the location and application of the equipment. Local peripheral equipment refers to that equipment that is located within the host computer complex facility. The second category, remote batch stations, refers to the peripheral equipment located away from the host computer complex, thus requiring direct communication between the station and the host computer. The factors considered in the statement of these requirements include: the software load analysis reported in Section 5.2; the computer simulation activities reported in Section 5.4; and the results of the vendor survey presented in Reference 22.

5.5.4.1 Local Peripheral Equipment

Local peripheral equipment for the host computer complex shall consist of: card reader, card punch, line printer(s), plotter, interactive display terminals and mass storage devices. The requirements for these devices are based primarily on the simulation support programs comprising the batch job stream. The job step characteristics of the programs entering the host computer job stream from the local peripheral equipment shall include tape and disc file access, FORTRAN compilations, program library updates, job execution, and interactive operations from the display terminals.

The following summarizes the requirements for the host computer local peripheral equipment as discussed in Reference 20:

<u>Peripheral Equipment</u>	<u>Requirements</u>
Card Reader	1 physical unit 1100 cards per minute 80 columns per card
Line Printer	3 physical units 3500 lines per minute total 120 characters per line
Card Punch	1 physical unit 100 cards per minute 80 columns per card

C

FINAL REPORT

DEVELOPMENT OF SIMULATION COMPUTER COMPLEX SPECIFICATION

MDC E0857
29 JUNE 1973

Magnetic Tape Units	5 physical units 7 tracks 560 and 800 bits per inch density 12K words per second transfer rate
Mass Storage Units	4 physical units 42 million words storage capacity 200K to 800K words per second transfer rate 30 ms average access time maximum 10 ms average latency maximum
	2 physical units 18 million words storage capacity 50K to 200K words per second transfer rate 90 ms average access time maximum 20 ms average latency maximum
Interactive Display Terminals	6 units Typewriter character set 6 special purpose function keys 8 x 10 inch CRT size maximum 75 feet max. communication distance
Plotter	1 unit (off-line) hardcopy plots of magnetic tape stored data extended periods of unattended operation chart size 8 1/2 x 11 inches maximum of 50 charts continuously

This peripheral configuration includes the reliability considerations noted in Section 5.5.5.

5.5.4.2 Remote Batch Station Peripheral Equipment

The host computer shall support the simultaneous processing load for two remote batch stations. One station is assumed to be located at the simulator development sub-contractor's off-site facility while the other station is assumed to be located on the NASA facility external to the computer complex. The job step characteristics of the programs entering the host computer complex job stream from a remote batch station shall include tape and disc file access at the host computer, FORTRAN compilations, program library updates, and job execution. Preset user directives supplied with the remote batch station job shall specify the host computer operations to be performed.

The following minimum peripheral equipment requirements are recommended for each remote batch station of the Training Simulation Computer Complex as derived in Reference 20:

<u>Peripheral Equipment</u>	<u>Requirement</u>
Card Reader	1100 cards per minute 80 columns per card
Line Printer	900 lines per minute 120 characters per line 48 character set
Card Punch	100 cards per minute 80 columns per card

The remote batch stations shall be capable of communication with the host computer complex at a transfer rate of 50K bits per second with a maximum communication distance of 10,000 feet.

5.5.5 Reliability

The reliability requirements for the computer equipment associated with the TSCC are of two distinct types: those which impose computer configuration requirements; and those which define associated component reliability requirements. The following paragraphs summarize the requirements for both.

Since the peripheral equipment of the computer system of the TSCC supports a variety of functions relating to the real-time simulation process and the non-real-time batch processing, a failure in any of the equipment could have varying degrees of severity. For example, a failure of a disc unit supporting the simulation job would cause the simulation session to end while the system was reconfigured to make other disc storage space available. Thus, although the computer system was actually capable of computing, it would have to be considered "down" until the real-time job was able to be continued. If in reconfiguring the disc storage in order to alleviate this problem, the non-real-time background batch could not be continued, the computer system would again be considered inoperative. In order to achieve the desired reliability for the computer system, problems similar to those described in the example above can be avoided by placing additional requirements on the configuration of the computer equipment. The computer configuration requirements which arise from these reliability considerations are discussed in Reference 20 and reflected in the peripheral configuration, Section 5.5.4 and the configuration summary.

DEVELOPMENT OF SIMULATION COMPUTER COMPLEX SPECIFICATION

The host computer central processing unit, memory units, input/output channels and input/output control devices must be of sufficient reliability to guarantee that the computer system will be available for simulation purposes for an average of 95% of the scheduled usage periods. These usage periods may extend up to twenty hours of the twenty-four hours of the day. This total system availability requirement is based upon the system configuration as described above.

5.5.6 Maintainability

The computer equipment must be capable of self-test using diagnostic routines. These routines will be capable of fault isolation to the replacement module. Provisions are required for monitoring the contents of memory locations and operational registers. Also, single step of instructions is required. Provisions are required for storage of these routines for quick access such as on disc files. These diagnostic routines are required to be controlled from an appropriate man-machine interface such as a control and display console or maintenance console for rapid interrogation.

Diagnostic routines are required for all peripheral equipment. These routines must isolate the failed unit and type of failure. For all nonessential peripheral units, noninterference maintenance will be required. A nonessential peripheral unit is defined as one where redundant units exist such as line printers, magnetic tape transports, etc., or any peripheral not needed for a mode of operation of the system complex such as remote stations, CRT terminal, card punch, etc. The computer system must be capable of performing tasks not requiring these peripheral units while maintenance is being performed.

The equipment must be capable of being repaired by replacement of easily removable circuit modules, cards, or components, whenever practical. Test points, adjustment points, cables, connectors, and labels will be accessible and visible during maintenance. Structural members or other mounted components of the unit or chassis will not interfere with access to components, connectors, or connections. Access doors, panels, or covers will open without interference for removal of items, whenever practical. Guards and shields will be provided, as necessary, to protect personnel and equipment from electrical voltages and interference.

DEVELOPMENT OF SIMULATION COMPUTER COMPLEX SPECIFICATION

Labels will be provided to identify and locate internal test points, adjustment points, connectors, and cables. All parts, subunits, and units likely to be replaced will be marked for identification. Marking will correspond with part numbers and reference designations used in assembly drawings and schematics. All parts, subassemblies, and assemblies will be interchangeable with other parts, subassemblies having the same part number. The equipment will be designed for a useful life of ten years of cyclic operation when supported by maintenance.

5.5.7 Computer Compler Expansion Capability

In the lifetime of the SMS, changing vehicle design, mission design or crew training goals may dictate that the computer hardware of the TSCC be expanded in order to provide the desired level of support. The following expansion requirements shall be met for the various computer hardware elements. Processing capability of the computer must be able to be expanded by either the addition of processing units to the existing computer or by upgrading the existing processor(s) to higher performance equipment. The memory capacity of the computer must be able to be expanded by the simple addition of memory blocks. The number of input/output channels must also be expandable in order to service an increased number of simulator devices or computer peripheral equipment. The computer channel capacity shall be expandable to accommodate two SMS simulators. Any modification to the computer equipment required to enable necessary expansion shall be able to be performed at the TSCC. Each vendor must indicate his expansion capabilities for the above requirements, and the amount of facility downtime required to implement these field changes. Additional expansion capabilities that may be provided by the vendor above the requirements set forth in this report shall be indicated.

5.5.8 Acceptance Test and Control Procedures

Acceptance testing is required for two specific purposes. The first requirement is to assure that the hardware delivered is functioning properly after shipping and installation. Test procedures, test software, test equipment and appropriately trained personnel are required for this testing and must be supplied by the computer vendor. Most, if not all, of this testing will be repeated periodically for maintenance purposes.

The second requirement for acceptance testing is performance demonstration. A performance demonstration by the vendor using customer supplied software should be

DEVELOPMENT OF SIMULATION COMPUTER COMPLEX SPECIFICATION

an acceptance test requirement. The most appropriate performance demonstration for the computer complex requirement is a benchmark program to demonstrate processor speed. An even better demonstration would be a synthetic simulation program requiring the memory and data flow or input/output rates associated with the simulation to be implemented. If this benchmark program is written in FORTRAN, the anticipated source language for the simulation software, then the benchmark demonstration would reflect the influence of such factors as compiler efficiency and compatibility, machine language instruction repertoire, processor speed, operating system overhead requirements, and channel insufficiencies.

An additional practice which should be considered is demonstration by the vendor of a substantial number of required existing applications software programs. These programs are supplied by the customer, converted by the vendor where necessary, and executed on the computer as a prerequisite to acceptance. This practice affords a number of highly desirable benefits. First, it forces the vendor to assume responsibility for software conversions which are due to his incompatibility with prior systems. Secondly, it creates a library of executable programs in hand on the date of acceptance. Thirdly, if the program executions are timed, it provides a measure of processor performance for a large sample of software. The numbers of programs executed for acceptance tests often run as high as 100-200 separate programs.

5.5.9 Environmental Constraints

The vendor must provide power generating or regulating devices for any power other than available facility power. Facility power is limited to the following (assuming a Building 5 installation):

208/120 VAC $\pm 5\%$, 3 phase/1 phase, 60 Hz ± 0.5 Hz

480/277 VAC $\pm 5\%$, 3 phase/1 phase, 60 Hz ± 0.5 Hz

The equipment must not be permanently damaged by electrical power failures and electrical transients to $\pm 25\%$ of nominal voltage. The equipment not be permanently damaged when subject to a non-operative temperature of -10° to $+60^{\circ}\text{C}$.

The vendor proposals must include facility requirements in sufficient detail to allow the NASA to evaluate the facility requirements. The following requirements must be provided by the vendor: outline drawings and physical dimensions, air conditioning requirements, weight and floor loading, power requirements, clearance requirements,

DEVELOPMENT OF SIMULATION COMPUTER COMPLEX SPECIFICATION

humidity and temperature limitations, grounding requirements, discussion of all considerations affecting preparation of a suitable installation site and means of facilitating installation, and restrictions or interconnecting cable lengths to all subsystems and peripheral devices. In addition, the vendor should define site preparation support to be provided by their site preparation personnel.

5.5.10 Inter-Computer Communications

At the present time the SMS computer system is required to communicate with only one other computer complex, the Mission Control Center flight support computers. The communication path is a digital serial special purpose line with channel adapters at both computers. The adapter at the TSCC will be government furnished equipment. Therefore, other than providing the channel to service this data path, no special requirements are necessary for this inter-computer communications path.

A second inter-computer communications path will exist for those vendors who propose a multiple computer configuration which does not share a common memory bank. This path, or paths if more than two computers are required, are the communication paths between each computer and every other computer of the proposed configuration. The data base or COMMON block data used and generated by each computer is required to be transferred to each other computer at the start and end of each processing frame. The vendor is required to estimate the computer load due to additional software processing and operating system functions to support these additional data transfers. The additional I/O load must also be estimated. The proposed multiple computer configuration must be capable of supporting the processing and I/O loads defined in Section 5.2 plus the additional loads discussed above.

5.5.11 Simulator Interface/Computer Compatibility

The simulator interfaces to the host computer channels are the responsibility of the data conversion equipment vendors and are described in Section 6. The Mission Control Center interface is not well enough defined at this time to adequately define requirements. A computer output channel is provided with the capability of transmitting at the maximum conceived rate of 256K bits per second.

5.5.12 Computer Performance Measuring Equipment

The usage of computer performance measuring equipment has been assessed. The

DEVELOPMENT OF SIMULATION COMPUTER COMPLEX SPECIFICATION

equipment considered was the class of hardware monitors which attach to selected test points (bit indicators) in the computer system and sample those bits at a prescribed rate. The data is generally collected on magnetic tape over a period of time determined by the nature of the test (typically from one day to two weeks). At the completion of the test period, the tape is processed by software programs supplied by the vendor of the hardware monitor, and a series of reports are generated which summarize the results of an analysis of the data. The reports generally contain tabular and graphical summaries of the data. The probes of the hardware monitor may, typically, be attached to any bit indicator of interest. Thus the user of the device has freedom in selecting the data to be collected in order to achieve a particular test objective.

The reports generated by the standard support software are generally merely summaries of the time history of bit changes. Very little interpretation of the results is provided. Quite often solutions to throughput and configuration problems are obtained only after careful study of the results by trained analysts.

A hardware monitor of this type is being used in the commercial computing facility of the McDonnell Douglas Automation Company in St. Louis. The unit is the X-RAY system by Tesdata. The unit has 48 probes which can sample data as often as every 10 μ seconds. The data collection process is controlled by an internal clock and the data is stored on magnetic tape.

A hardware monitor can be an important asset to a commercial computing facility in which the highest possible utilization of resources is essential, especially in the face of an ever changing job mix. For the TSCC, in which the job mix is well defined and unchanging, the utility of such a device is lessened. In addition, the requirements for a hardware monitor depend to a minor degree on the computer equipment to be maintained. Since the computer vendor for the TSCC will not be known in the immediate future, it is recommended that a hardware computer performance measuring device not be required at this time.

5.5.13 Minimum Configuration Requirements

Table 5-6 summarizes the resulting minimum configuration requirements for the computer hardware. Besides the equipment requirement, a cross-reference to the appropriate report paragraphs is given. The referenced paragraphs may be consulted for a more detailed statement of requirements for a specific unit.

Table 5-6
Computer Hardware Minimum Configuration Requirements

EQUIPMENT	REQUIREMENT SUMMARY	PARAGRAPH REFERENCE
CPU	1 CPU 8.4 MIPS 2.1 MOPS 2 CPU 5.0 MIPS 1.25 MOPS 4 CPU 3.1 MIPS 0.78 MOPS	5.5.1
Memory Words	1 Computer 284000 2 Computer 213000 4 Computer 184000	5.5.2 through 5.5.11
I/O Channels	Channel Capacity Sufficient to Support the Following Devices at the Specified Loading: 6 Devices @ 500K WPS 2 Devices @ 200K WPS 1 Device @ 90K BPS	5.5.3
Local Peripheral Equipment	1 High Speed Card Reader 3 High Speed Line Printers 1 Card Punch 5 Magnetic Tape Transports 4 Fast Access Disc Devices 2 Medium Access Disc Devices 6 Interactive Display Devices 1 Plotter	5.5.4, 5.5.5
Remote Peripheral Equipment	2 Remote Batch Stations, each consisting of: 1 High Speed Card Reader 1 High Speed Line Printer 1 Card Punch	5.5.4, 5.5.5 through 5.5.11

DEVELOPMENT OF SIMULATION COMPUTER COMPLEX SPECIFICATION

5.5.14 Requirements Impact for Two Simulators

The requirements presented in the previous sections have defined the computer hardware equipment required to support one Shuttle Mission Simulator. If the computer complex will be required to support two simulators, the requirements in this section must be adjusted to account for several items.

If the two simulators are to be of identical fidelity and capability, then the computer hardware requirements must, in general, be doubled. The CPU(s) for example, must be capable of simultaneously supporting the worst case loading for each computer. In single computer, single CPU systems, consideration must be given to the possibility of overlapping the I/O, thus utilizing the CPU nearly 100% for simulation processing and reducing slightly the requirement from double the amount stated above. Multiple CPU or multiple computer configurations will also require double the requirements stated above. In all single computer configurations, system software overhead requirements may become highly significant due to the increased activity of allocating resources to two simulation jobs. CPU requirements must be carefully considered in this area.

Memory requirements will also be required in twice the amount of the requirements stated above.

The I/O loading would also be multiplied by two. I/O channel configuration requirements will have to consider simultaneous, asynchronous communication to multiple sets of the simulation hardware interfaces.

The requirements for computer peripheral equipment do not necessarily have to be increased. The software development load would remain about the same for two simulators, since the same software modules would be used in each simulator. Hence, the amount of peripheral equipment would not have to be doubled. The exception to this are those peripheral devices which directly support the simulators. Multiple units would be required in those areas. Multiple computer complexes may also force an increase in peripheral equipment requirements. If the peripheral equipment cannot be shared between computers, then each computer would be required to have an adequate set of peripheral devices.

DEVELOPMENT OF SIMULATION COMPUTER COMPLEX SPECIFICATION
Section 6

DATA CONVERSION EQUIPMENT REQUIREMENTS - TASK 1.3

The data conversion equipment is defined as that equipment between the host computer and the operational hardware. This data conversion equipment includes minicomputers, interfaces to the various computers, and digital-to-analog, analog-to-digital, and digital-to-digital conversion equipment. For the simulator configuration, shown in Figure 3-1, the data conversion equipment provides data paths between the host computer and the simulator, the host computer and the flight computers, the host computer and the Mission Control Center, and the host computer and the instructor displays.

There is data conversion hardware associated with the computer and the simulator hardware that is usually procured from the vendors of these systems. Where appropriate, the dividing line is delineated and discussed herein. Particularly, that equipment considered as part of the simulator hardware is noted. The options available for procurement of the data conversion equipment are the following: procurement as part of the host computer complex, procurement as part of the simulator equipment, or procurement as a separate package.

This section presents the functional requirements and assumptions, and the hardware requirements for the data conversion equipment of one simulation complex. For two simulators of equal complexity, two identical sets of data conversion equipment are required. However, since the functional requirements and assumptions are based upon preliminary data, these may require revision as the Shuttle design evolves, and the simulator requirements become firm. Consequently, tailoring of the hardware requirements presented herein may be required. For instance, each of the redundant flight computers has its own set of data conversion equipment. Reductions in the flight computer redundancy for a simulator reduces the number of sets of data conversion equipment required. Elimination of the motion base has a minimal effect as presented herein; whereas, using a television camera and model system instead of Computer Image Generation for the payload handling visual scenes has a significant impact by increasing greatly the number of analog outputs. The impact of other additions or deletions to the simulator crew station may be readily estimated using the data presented herein.

The results of the Data Conversion Equipment Requirements Task 1.3 have been presented in detail in Reference 20. This Section summarizes the results of that Task, however, much of the rationale and analysis are omitted.

DEVELOPMENT OF SIMULATION COMPUTER COMPLEX SPECIFICATION

6.1 HOST COMPUTER TO SIMULATOR DCE

Data conversion equipment is required between the host computer and the crew stations, motion base systems, visual systems, and aural systems. The data conversion equipment, as shown in Figure 6-1, includes: digital-to-analog, analog-to-digital, and digital-to-digital conversion equipment; a minicomputer; and interfaces between the minicomputer and host computer, between the minicomputer and conversion equipment, and between the minicomputer and the payload handling visual generation computer. Also included is the real-time clock for simulation timing.

The data conversion equipment, as defined, assumes that the digital-to-analog converters and analog-to-digital converters are located at the minicomputer. The digital inputs and outputs are assumed to be distributed between the simulator and the data conversion equipment with that portion in the simulator not specified.

The data conversion equipment transmits or receives a 16-bit word and outputs an accompanying 11-bit address. The simulator equipment decodes the 11-bit address and provides the storage registers, line receivers, and lamp and flag drivers for receiving data, and the digital multiplexer and line drivers for transmitting data. The specifications provide for interfacing to 16 separate units of simulator equipment.

By transmitting the digital data as a 16-bit word and 11-bit address, large numbers of cables with line drivers and receivers are eliminated between the data conversion equipment and the simulator. If the digital storage were located external to the simulator, a separate wire plus a line driver and receiver would be required for each discrete bit. By using integrated circuits the burden placed upon the simulator in weight and space is not great for the decoders and storage. However, additional power supplies will be required at the simulator.

The digital-to-analog and analog-to-digital converters are located external to the simulator. The number of cables for these inputs and outputs is not great. Also, the size of these units tend to be greater than for the discretes. In addition, since one reference supply and one analog-to-digital converters is preferable, these are accommodated more easily at a central location especially if several separately located units of simulator equipment are configured.

6.1.1 Minicomputer

A minicomputer is required to function as a control processor between the data conversion equipment and the host computer.

It is assumed that floating point to fixed point or fixed point to floating conversion will be performed in the host computer. The data words will be transmitted to/from the minicomputer will be in fixed point format scaled to less than 1.0. Therefore, the software program requirements are reduced to a simple executor for data block definition and a data transfer algorithm. In addition, the minicomputer controls the real-time clock period and delay from data received from the host computer. The minicomputer also processes interrupts from the real-time clock.

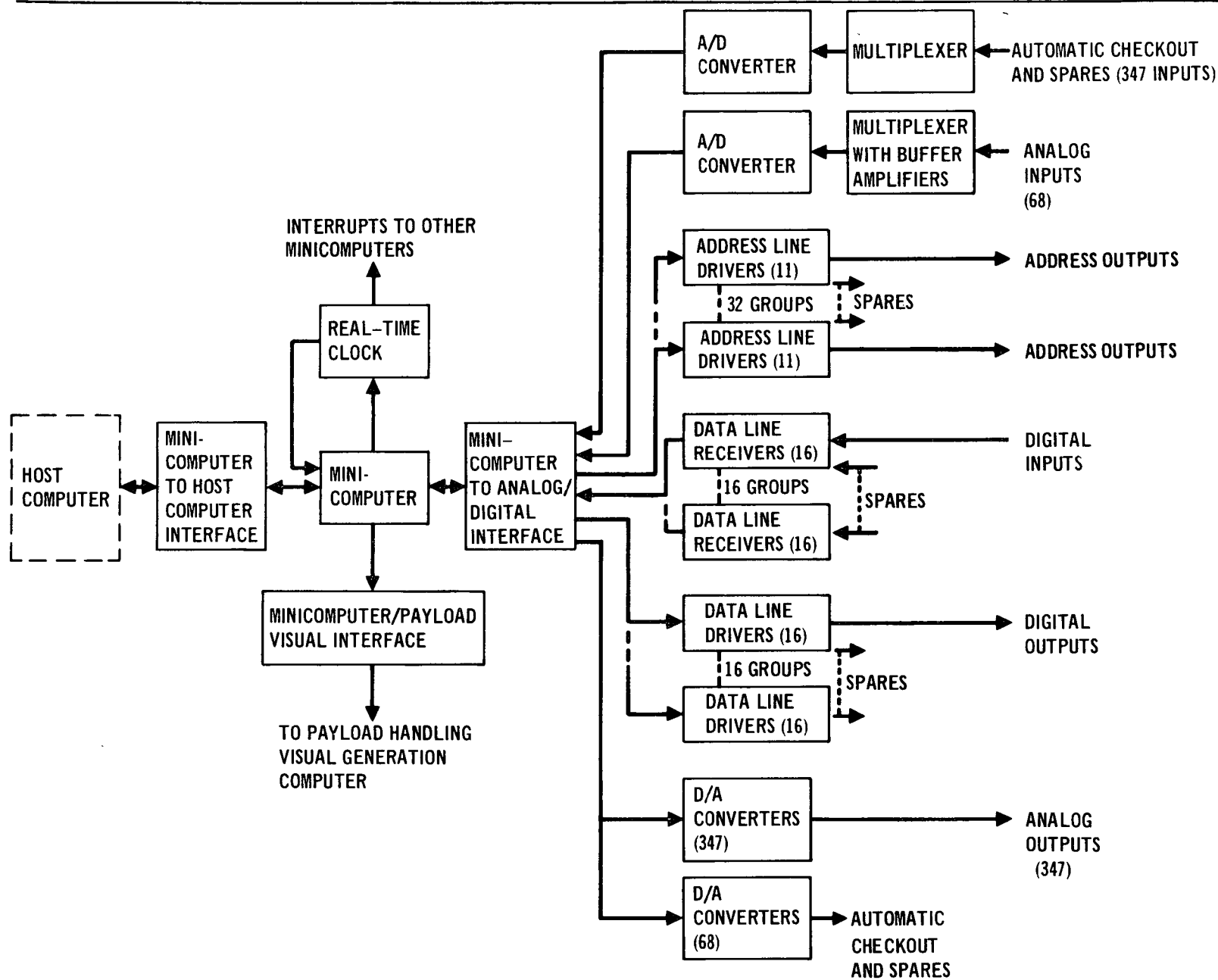


Figure 6-1 Host Computer to Simulator DCE

DEVELOPMENT OF SIMULATION COMPUTER COMPLEX SPECIFICATION

The memory will be used primarily for buffer storage of data. Data will be buffered into dedicated memory areas reserved for host computer data and simulator data. The addresses in the buffers will be assigned according to the sequence of data transfer. Core memory required is 8,192 16 Bit.

The 16 Bit word size is compatible with the most stringent analog and resolver accuracy and resolution requirements. A memory cycle time of a maximum of 1.0 microseconds is required.

6.1.1.1 Simulator Crew Station Input/Output

The simulator data is transmitted at three different rates: 25 times per second, 12 times per second, and 6 times per second. However, for worst case it is assumed peak loading can occur where all the data is transmitted during one 40 millisecond time frame. Since it is desirable that the transfer occur during a quiescent time, the portion of the frame devoted to input/output to the simulator is assumed to be 50%.

In summary, the resulting data transfer rates for each type of transfer to the simulator crew station are as follows:

Analog input - 20,000 words per second minimum
Analog output - 100,000 words per second minimum
Digital input - 100,000 words per second minimum
Digital output - 100,000 words per second minimum
(one word = 16 bits)

6.1.1.2 Host Computer Input/Output

To minimize the portion of the frame time required to transfer data to the host computer, it is assumed that 10% of the frame time will be used. Thus, transmitting the data in 10% of the frame time results in a data transfer rate to the host computer of 224,000 words/second or rounded to 200K words/second. Each word is specified at 16 bits.

6.1.1.3 Payload Handling Visual Input/Output

A separate data channel in the minicomputer is required to interface the special purpose, image generation computer for the payload handling scenes. The amount of data transmitted is estimated, at 256 words per frame. Assuming the data is transmitted in 5% of the total frame time to minimize interference with the other operations results in a data transfer rate of 128,000 words per second.

Additional Requirements

The peripheral equipment will include a paper tape punch with a speed of not less than 50 characters per second and paper tape reader with a speed of not less than 300 characters per second for maintenance and off-line checkout.

6.1.2 Real-Time Clock

The real-time clock generates two output pulses: a pulse to determine the start of each simulation frame and a delayed pulse after the frame pulse. The period of the frame sync pulse and the amount of delay of the delayed pulse is required to be programmable from the minicomputer. The delayed pulse will interrupt the host computer through the minicomputer to host computer interface. The frame sync will interrupt seven separate minicomputers. Line drivers are required at the output, to drive the signals a maximum distance of 100 feet. Also, hold, reset, and start control from the minicomputer is required. This capability allows the simulation to be initialized, started on request, and stopped and restarted during the real-time operation.

In addition, the real-time clock is required to have the following minimum characteristics:

Accuracy, minimum	.01% of value
Resolution	1.0 millisecond
Programmable period	0.001 to 1 second
Programmable delay	0.001 to 1 second

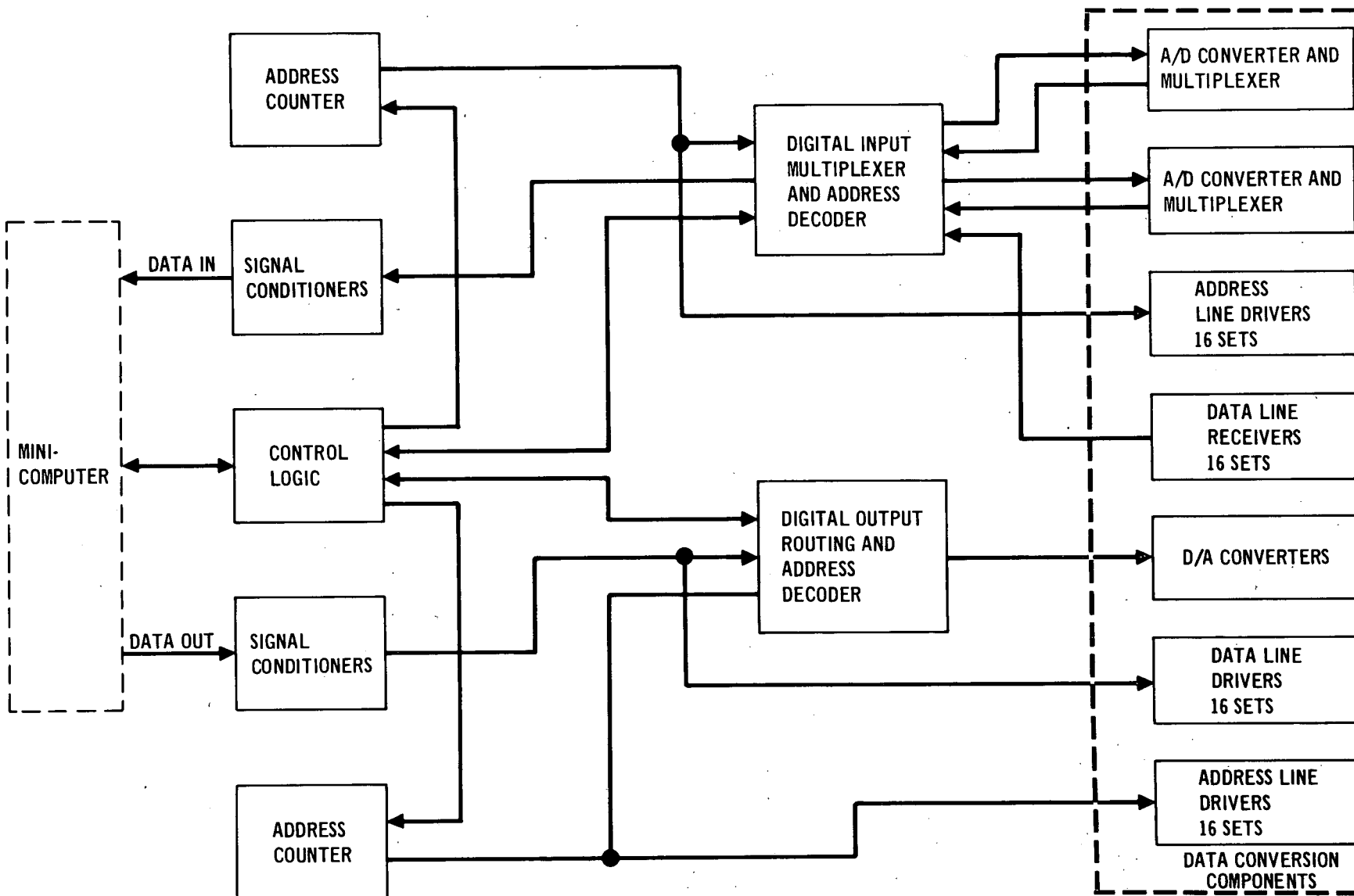


Figure 6-2 MiniComputer to Analog/Digital Interface

6.1.3 Minicomputer to Analog/Digital Interface

An interface shown in Figure 6-2 is required to transmit digital data between the minicomputer and the data conversion components. The data will be transmitted in blocks to/from a minicomputer direct access channel. The minicomputer will determine the amount of data in the block and will initiate each block transfer.

Addresses must be generated from the data sequence. One method is to have two address counters - one for input data and one for output data. To adequately address all components and provide for sufficient spares, 11-bit addresses are specified. The 11-bit address from one counter will be transmitted concurrently with each data word to the digital output or will be decoded for selecting the recipient digital-to-analog converter. The address which accompanies the digital data will be decoded at the simulator crew station.

For those systems so constructed, alternate methods of addressing may be used. One other method is to output an 11-bit address from the minicomputer for each data word thus negating the need for address counters in the interface. This method increases the memory size and input/output transfer rate requirements of the minicomputer. The vendors may provide alternate methods so long as equivalent capability is provided.

The interface will provide 16 parallel address outputs and 16 digital output words for cabling to equipment at different locations. The same address and digital words will appear in each set, i.e., the sets are parallel. This will allow equipment at 16 different locations to be cabled to the interface. There is no restriction as to the division of components among equipment, i.e., all possible components using all allowable addresses could be connected to one set of outputs. Also, the interface will provide for transmitting addresses and receiving digital words from 16 different equipment sources. The digital word inputs will be OR'd for routing to the minicomputer. The same address will appear in each set. As with the digital output words, there is no restriction to the division of addressable components. The sole purpose of the additional parallel outputs and inputs is for ease of cabling.

The characteristics of the various control and timing lines are a function of the specific minicomputer and data conversion components.

6.1.4 Minicomputer to Host Computer Interface

An interface is required to transmit sixteen bits of parallel data in both directions between the host computer and minicomputer. Line drivers and line receivers are required at both the minicomputer and host computer. In addition, the interface receives two separate outputs, a frame sync and a delay, from the real-time clock. The interface interrupts the minicomputer upon receipt of the frame sync and interrupts the host computer upon receipt of the delayed output. All control logic to effect the data transfer and real-time clock interrupts, and to assemble and disassemble data words, as required, will be incorporated. The request for data transfer is controlled from the transmitting computer. The data transfer rate will be a minimum of 200K words per second, and the maximum distance of the data transfer will be 100 feet. Other characteristics of the interface such as voltage levels and timing are a function of the minicomputer and host computer.

6.1.5 Minicomputer/Payload Visual Interface

An interface is required to transmit 16-bit words in blocks to the payload handling visual generation computer from the minicomputer. The amount of data in the output block is controlled by the minicomputer. The data transfer rate required is a minimum of 128K words per second. Line drivers and line receivers are required at both the minicomputer and visual computer. All control logic to effect the data transfer is required. The maximum distance of the transfer will be 50 feet. Other characteristics of the interface such as voltage levels and timing are a function of the minicomputer and visual generating computer.

6.1.6 Data Conversion Components

The following lists the requirements for the digital-to-analog, analog-to-digital, digital-to-synchro/resolver, and digital-to-digital conversion equipment for data conversion and conditioning for transmission to the crew stations, motion base systems, visual systems, and aural systems.

Analog Inputs:

The requirements for the analog-to-digital conversion equipment for the simulator equipment are summarized as follows:

Analog Input System Requirements

Total system throughput rate, minimum	20,000 words per second
Resolution	12 bits including sign
Overall accuracy including multiplexer and buffer Amp	$\pm 0.25\%$ FS + 1/2 LSB
Input voltage range	0 to ± 10 VFS
No. of multiplexer input channels with amplifier, minimum	68
Input impedance	Greater than 1M ohm shunted by less than 200 pf
Crosstalk rejection	80db, DC-1Khz, 1K ohms source impedance

Analog inputs are also required to accomplish the automatic checkout.

To minimize interference with the simulator inputs, a separate analog-to-digital converter with multiplexer is specified. All the analog outputs will be monitored by the multiplexer; therefore, three hundred twenty 8-bit, six 12-bit, and twenty-one 14-bit inputs are required. This throughput may be accomplished with one or more analog-to-digital converter and multiplexer systems. The measurement system must have resolution and accuracy greater than the analog outputs to be measured; therefore, the resolution is specified at 15 bits including sign and the overall accuracy is specified at 0.025%. The system throughput rate need not be great; since the system checkout is not real-time dependent. The requirements are summarized as follows:

Analog Input Automatic Checkout System Requirements

Total system throughput rate, minimum	2,000 words per second
Resolution	15 bits including sign
Overall accuracy including multiplexer	$\pm 0.025\%$ FS $\pm 1/2$ LSB
Input voltage range	0 to ± 10 VFS
No. of multiplexer input channels, minimum	347
Input impedance	Greater than 100K ohms shunted by less than 200 pf
Cross talk rejection	80 db, DC-1 Khz, 1K ohms source impedance

The requirements for the digital-to-analog conversion equipment are summarized as follows:

8-Bit Digital-to-Analog Converters

Number required	320
Resolution	8 bits including sign
Accuracy	Greater than 0.5% FS \pm 1/2 LSB
Digital storage	Single buffered
Total system throughput rate, minimum	100,000 words/second
Settling time, maximum	100 microseconds to 1% of final value
Output characteristics	0 to \pm 10VFS, 10ma, less than 10 ohm output impedance

12-Bit Digital-to-Analog Converters

Number required	74
Resolution	12 bits including sign
Accuracy, minimum	Greater than 0.1% FS \pm 1/2 LSB
Digital storage	Single buffered
Total system throughput rate, minimum	100,000 words/second
Settling time, maximum	100 microseconds to 0.1% of final value
Output characteristics	0 to \pm 10VFS, 10ma, less than 10 ohm output impedance

14-Bit Digital-to-Analog Converters

Number required	21
Resolution	14 bits including sign
Accuracy	Greater than 0.05% FS \pm 1/2 LSB
Digital storage	Single buffered
Total system throughput rate, minimum	100,000 words/second
Settling time, maximum	100 microseconds to 0.1% of final value
Output characteristics	0 to \pm 10VFS, 10 ma, less than 10 ohm output impedance

The requirements for the digital inputs are summarized as follows:

Digital input	16 bits
Address output, minimum	11 bits
Source impedance	100 ohms nominal
Input cabling	#22 twisted pair
Input voltage range	Vendor dependent
System throughput rate per 16-bit word, minimum	100,000 words/second
Line drivers, quantity	176 (groups of 11)
Line receivers, quantity	256 (groups of 16)

Digital Outputs

The requirements for the digital outputs are summarized as follows:

Data output	16 bits
Address output, minimum	11 bits
Source impedance	100 ohms
Output cabling	#22 twisted pair
Output voltage range	Vendor dependent
System throughput rate per 16-bit word, minimum	100,000 words/second
Data line drivers	256 (groups of 16)
Address line drivers	176 (groups of 11)

Digital-to-Synchro/Resolver Conversion

The requirements for the digital-to-synchro/resolvers converters are summarized as follows:

Number required	18
Resolution	10 bits
Accuracy	± 16 minutes $\pm 1/2$ LSB
Reference	26V, 400 Hz
Number required	90
Resolution	14 bits
Accuracy	± 6 minutes $\pm 1/2$ LSB
Reference	26V, 400 Hz

FINAL REPORT

DEVELOPMENT OF SIMULATION COMPUTER COMPLEX SPECIFICATION

MDC E0857
29 JUNE 1973

Number required	15
Resolution	16 bits
Accuracy	± 1 minute $\pm 1/2$ LSB
Reference	26V, 400 Hz

6.2 HOST COMPUTER TO FLIGHT COMPUTER DCE

The equipment proposed to perform the data conversion and transfers between the host computer and individual flight computers consists of a microprogrammable minicomputer and two specially designed direct memory access interfaces. One minicomputer is required per flight computer because of the high I/O rates and large processing demands.

6.2.1 General Minicomputer Requirements

In view of the large amount of data processing to be performed by the minicomputer, as much of the work as possible should be accomplished by the microprogram. This includes control of all I/O or DMA transfers, reformatting of data words from host computer to flight computer format and vice versa, in either fixed or floating point, and any required interrupt servicing. In general, these operations can be performed 5 to 10 times faster by a microprogram than by a program stored in main memory.

Microprogram Requirements

It is desirable that the read-only memory be alterable to facilitate program development and to accommodate changes or additions. A minimum read-only memory word size of 32 bits allows a reasonable degree of parallelism in the microinstruction. Arithmetic operations, register shifts, conditional testing, memory cycle or I/O initiation, and data path manipulation can be simultaneously performed for increased processing speed. It may be worthwhile to note here that larger word sizes (i.e., 64 bits), while providing more parallelism, do appear to significantly increase the programming difficulty. This would in turn require a higher degree of proficiency and training on the part of the programmer.

The basic requirements for the microprogram control unit and read-only memory are summarized as follows:

Word size, minimum	32 bits
Cycle time, maximum	200 nanoseconds
Memory size, minimum	512 words
General purpose registers, minimum	6

DEVELOPMENT OF SIMULATION COMPUTER COMPLEX SPECIFICATION

Central Processor and Core Memory Requirements

The core memory section of the processor is used primarily for storage of data and addresses to be operated on by the microprogram. Figure 6-3 shows the different memory blocks and their relationship with the rest of the system. Data and address outputs from the flight computer are buffered into a dedicated memory area, and then reformatted and/or transferred on a word by word basis to another memory section reserved for DMA transfer to the host computer. Outputs from the host computer are buffered into a dedicated memory area, and then reformatted, as required, and stored in a data array according to flight hardware or intercomputer address. Flight hardware data transfers to the flight computer are initiated by an address word from the flight computer related to a specific location or locations in the data array. Intercomputer transfers to the flight computer are initiated by the minicomputer.

The required memory size is calculated as follows:

Host computer I/O buffers (2400 - 32 bit words x 2)	4800 words
Flight computer data array (1800 - 32 bit words x 2)	3600
Flight computer output buffer (1200 - 32 bit words x 2)	2400
Address array	1000
Peripheral driver routines and spares	<u>4200</u>
Total	16000 words

A word size of 16 bits is specified because it is compatible with the I/O characteristics of the flight computer and is a standard format in the minicomputer industry. It also allows single word addressing in the minicomputer for minimum memory access times.

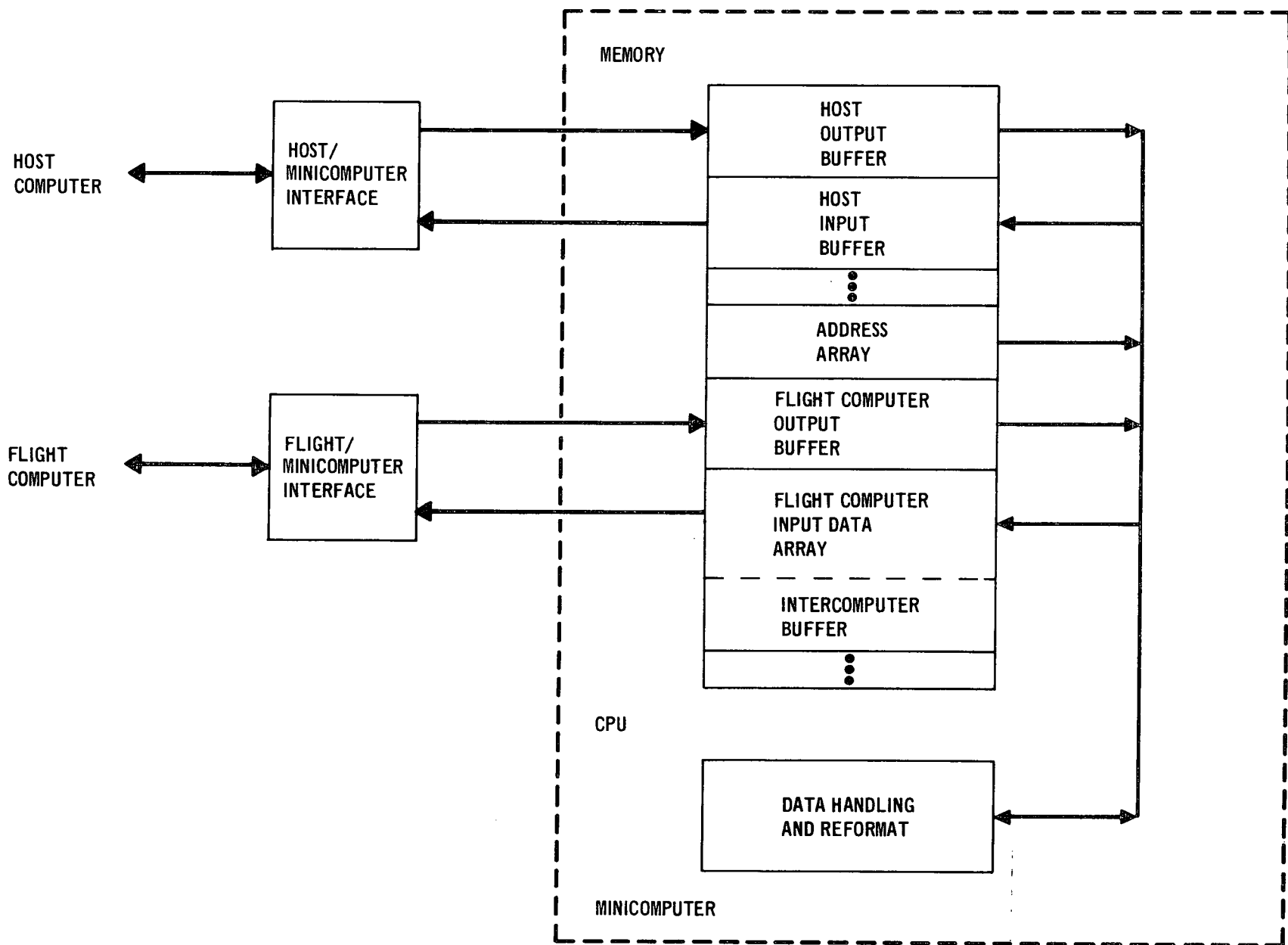


Figure 6-3 Host Computer to Flight Computer DCE

DEVELOPMENT OF SIMULATION COMPUTER COMPLEX SPECIFICATION

A 1.0 microsecond cycle time is required to accommodate the DMA transfer rates to and from the flight computer, and for the large number of memory to memory transfers which take place during each frame.

Six general purpose registers are required in order to allow sufficient scratch storage for efficient microprogram operation. These, of course, are in addition to the required I/O and memory registers.

I/O Characteristics

The minicomputer will interface two completely asynchronous computers operating at 40 millisecond frame rates. Since the data transfers are independent, the minicomputer must establish memory access priority levels for the various types of transfers. The data handling structure is shown in Figure 6-3 for this interface.

The host computer I/O is defined as a 32-bit or greater parallel word output with the most significant 32 bits transmitted. This data will be block transmitted each frame, an input and output block. The flight computer I/O is defined as a program controlled or DMA transfer of 16-bit parallel words. It is assumed that flight hardware transfers are under program control and that intercomputer transfers are under DMA control.

All data transfers with the minicomputer are assumed to be DMA because of the high I/O rates involved. The characteristics of most DMA channels are such that a priority structure is needed to avoid access conflicts. The real-time program of the flight computer requires that it have the highest DMA priority. All other memory access activity should defer to flight computer data transfers. The next level of priority is assigned to the host computer. The floating-point reformat and inter-buffer relocation activity is, therefore, relegated to a third priority in that access to the buffer memory is not possible during DMA transfers unless a minicomputer with a dual-port memory architecture is selected.

The basic I/O structure requires an interface to service the host computer and one to service the flight computer. These interfaces are unique and require some development but should not be too different from existing DMA block transfer controllers available with most minicomputers. The operation of these two interfaces is described in subsequent paragraphs.

DEVELOPMENT OF SIMULATION COMPUTER COMPLEX SPECIFICATION

All flight computer output data is transmitted to the flight computer output buffer, shown in Figure 6-3. The minicomputer accepts the data at real-time rates via the DMA interface. The buffer is loaded sequentially with interleaved address and data words. Then the data is reformatted and/or relocated into the host computer input buffer as fast as possible within the given priority constraints. Data and address words are transmitted to the host input buffer in the same sequence as they originate from the flight computer. An interrupt from the real-time clock, described in Section 6.1.2, temporarily stops the reformat process and block transfers all data in the host input buffer to the host computer. Subsequently, the reformat process continues and reloads the input buffer. In order to avoid the partial transfer of a block of data, it is necessary to delay the host data transfer while a data block is being reformatted. In this case, a data block is defined as all the data words associated with one address word. This arrangement will appreciably delay the host data transfer if the minicomputer is in the process of reformatting a large block of data such as an intercomputer transfer.

New data for the flight computers is block transmitted from the host computer and temporarily stored in the host output buffer, shown in Figure 6-3. When the block transfer is complete, the minicomputer must reformat and/or relocate the data into the proper locations of the flight computer input array. Address words in the host data block are used to determine the proper storage locations for the data. An exception to this is the address words that identify the intercomputer data blocks sent to the flight computer. These addresses must be retained to initiate DMA transfers.

The two types of data to the flight computer may require different handling techniques. Assuming that flight hardware data is interrogated under flight computer program control, the flight computer can, by outputting a read-data address word, initialize the DMA interface to obtain the data word memory location from the address array. This memory location can then be used to obtain the data for the flight computer. In order to accommodate the transfer rates, all data parameters must be stored in fixed locations, and the operating rate of the minicomputer's DMA channel (including latency time and two DMA transfers) must be faster than 4 microseconds per address word access. Most minicomputers with a 1 Mhz DMA channel will operate at this speed.

DEVELOPMENT OF SIMULATION COMPUTER COMPLEX SPECIFICATION

Intercomputer transfers present a different situation in that an intercomputer address command causes data to be extracted from one flight computer, transmitted through the host computer, and then inserted into the requesting flight computer. This transmission sequence will cause a time lag between a transfer request and the data response that could easily be as long as 3 simulation cycles (120 milliseconds). The avoidance of such a frame lag, if intolerable, can be difficult. The most direct approach is to store in each minicomputer all of the intercomputer data that might normally be accessed by the associated flight computers. However, if a large block of data is required, the update task quickly becomes impractical. An alternate method is available if the occurrence of intercomputer transfers can be predetermined. Based on the flight computer operational software programs, those external events that effect intercomputer transfers will be known by and under control of the simulation program. This foreknowledge can permit the host computer to prepare for an intercomputer transfer before a flight computer request is generated. For example, the host computer knows that a flight computer, receiving out-of-tolerance data from the simulation, will request corroborative data from another computer. The host can transmit the appropriate data to the minicomputer before the request is made. Another alternative is to transfer data directly between flight computers through hardware interfaces. This method minimizes the time lag; however, the flexibility of introducing failures from the host computer is eliminated and the complexity of the interface is increased.

To support DMA transfers to the flight computer, the minicomputer's DMA channel must be capable of 1M words/second. Since the two machines are not synchronous, there will be some delay in a DMA transfer initialization. Also, updates from the host computer will not be in sync with flight computer requests, and one frame data lags are unavoidable.

6.2.2 Host/Minicomputer Interface

This interface connects the second highest priority DMA channel of the minicomputer with the host I/O channel. Data transfers are initiated by the transmitting computer, and the transmitting computer controls the amount of data. The interface receives the frame sync output from the real-time clock, and interrupts the minicomputer. Upon receipt of this interrupt, the minicomputer initiates data transfers to the host computer. The host computer may initiate data transfers to the minicomputer at any time during the frame.

All transfers will be block transfers; the host I/O rate is one-half the minicomputer rate or 500K words/second. The interface accepts one 32-bit word from the host and performs two 16-bit DMA transfers to the minicomputer. If the host computer word is greater than 32 bits, the remaining bits will be dropped. The reverse sequence is required for block transfers to the host. The maximum distance of the data transfer is 50 feet.

The host reads from or writes to the same buffer locations; therefore, the starting address for the interface will always be the first location in one of the two host data buffers. The interface must have the means to inform the receiving computer (host or minicomputer) when data transfers are complete. This interface can be very similar to existing DMA block transfer controllers that are available with most minicomputers. Other characteristics of the interface such as voltage levels and timing are a function of the minicomputer and host computer. The minicomputer vendor will supply the interface as part of the total system. Some development may be required; however, the development risk is negligible because many similar systems have been implemented previously.

6.2.3 Flight/Minicomputer Interface

This interface connects the highest priority DMA channel of the minicomputer with the flight computer I/O channel. Data transfers will normally be initiated by the flight computer except for the case of the block transfers to the flight computer's DMA channel.

Data transfers may be single word or block transfers and the data rates may vary. The maximum data rate will be 1M words/second. The interface transfers 16-bit words to either computer.

Flight computer outputs are always to the same buffer. The current load address of that buffer can be used as the starting memory address during interface initialization for flight computer outputs. The flight computer address word is used to initialize the interface for inputs.

The basic operation of this interface is similar to normal DMA block transfer controllers. It should be noted that the unit must accommodate DMA transmissions

FINAL REPORT

DEVELOPMENT OF SIMULATION COMPUTER COMPLEX SPECIFICATION

MDC E0857
29 JUNE 1973

under control of either computer and must be capable of direct memory access to either computer. Other characteristics of the interface such as voltage levels and timing are a function of the minicomputer and flight computer. The minicomputer vendor will supply the interface as part of the total system. Some development will be required; however, the development risk is negligible because many similar systems have been implemented previously.

6.3 SIMULATOR FLIGHT GRAPHICS

The data conversion equipment proposed to perform the data transfers between the host computer and the flight graphics displays consists of a minicomputer and two specially designed input/output interfaces. The minicomputer must interface with six flight graphics displays.

6.3.1 Minicomputer Requirements

The primary function of the processor is to transfer display words from the host computer to the flight graphics system. Refresh is to be performed in the flight electronics; therefore, it is not a requirement for the minicomputer. Also, no data conversion is to be performed in the minicomputer; therefore, there is no requirement for fast arithmetic capability.

Central Processor and Core Memory Requirements

The word size of 16 bits is compatible with the I/O characteristics of the graphics system as well as being a standard for minicomputers. Also, a minimum cycle time of 1.0 microsecond is specified to accommodate the DMA transfers and buffer to buffer transfers of data.

Core memory size requirements are determined as follows:

Data Storage

(1000 words/CRT x 6 CRT's)	-	6000 words
I/O drivers and executive	-	1000
Loader	-	500
Peripheral driver routines and spares	-	<u>4500</u>
Total	-	12000 words

I/O Characteristics

It is assumed that only one CRT will be interrogated per frame. The 1000 words/frame transfer rate is assumed for one serial multiplex bus servicing all six CRT's. At 20 microseconds per word, this allows the data to the CRT's to be transmitted in 1/2 the frame time.

DEVELOPMENT OF SIMULATION COMPUTER COMPLEX SPECIFICATION

The assumptions require the host computer to transfer a maximum of 500 32-bit words per frame. Since the update of the display units is not real-time critical, the host can transmit an uninterrupted block of data once per frame and the displays can be serviced after the host transfer. The host transfer should be completed as quickly as possible to minimize the host computer I/O time and allow maximum time for minicomputer operations. Using an 1.0 millisecond host transfer of the 1000 16-bit data words results in a 1.0 MHz DMA transfer for the mini-computer.

Since reformatting of data is not required in this interface, only one memory buffer section is required in the minicomputer. This buffer has interleaved address and data words, and will be loaded and read sequentially every frame with the starting address at the beginning of the buffer servicing the recipient CRT.

The high data rates of the host I/O require an interface with the DMA channel of the minicomputer. The relatively slow I/O to the display units can be handled by minicomputer program control via a DMA interface or a buffered I/O unit. The main operational problem associated with this interface is that the minicomputer must locate and interpret the address words in the host computer output buffer in order to control the I/O with the display units. Frame skipping occurs, but frame skips are not a problem at the two times per second update rate anticipated from the host computer.

6.3.2 Host to Minicomputer Interface

This controller must interface the host computer with the minicomputer DMA channel. Data transfers are initiated by the host computer with the minicomputer the slave computer. The host computer may initiate data transfers at any time. All transfers will be block transfers with a 1 MHz minicomputer DMA channel. The host will transmit at 500K words/second. The controller accepts one 32-bit word from the host and performs two 16-bit DMA transfers to the minicomputer. If the host computer word is greater than 32 bits, the remaining bits are dropped.

6.3.3 Flight Graphics to Minicomputer Interface

This interface connects the display unit data bus with the minicomputer. The controller will send and receive words under the minicomputer's program control at 20 microsecond intervals. The interface consists of a standard DMA controller or

FINAL REPORT

DEVELOPMENT OF SIMULATION COMPUTER COMPLEX SPECIFICATION

MDC E0857
29 JUNE 1973

I/O buffer and a specially developed parallel/serial control and transfer unit. Address word information from the minicomputer is used to initialize the controller for the type of data and the buffer location associated with each transfer. Other characteristics such as voltage levels and timing are a function of the minicomputer. The minicomputer vendor will supply the interface as part of the total system. Development will be required; however, the development risk is negligible because many similar systems have been implemented previously.

DEVELOPMENT OF SIMULATION COMPUTER COMPLEX SPECIFICATION

6.4 GRAPHICS DISPLAY TERMINAL FOR INSTRUCTOR STATION

A graphics display terminal with six CRT displays and inter-active keyboards is required for the instructor station. The display terminals will include a minicomputer and an interface between the minicomputer and host computer. The display terminal will be capable of showing both alphanumeric and graphic information.

6.4.1 Minicomputer

A minicomputer is required to function as a control processor between the graphics displays, including keyboards, and the host computer. The minicomputer will format the data to/from the host computer and provide CRT refresh as required. The minicomputer will also store inputs from the keyboards and display these inputs on the CRT's for editing before being transmitted to the host computer. The minicomputer will store the total input message then perform a block transfer to the host computer. The real-time clock will interrupt the minicomputer each frame. The minicomputer will output keyboard data to the host computer upon receipt of this real-time interrupt. The transfer of data from the minicomputer to the host computer is controlled from the minicomputer, and the transfer of data from the host computer to the minicomputer is controlled from the host computer. The computer receiving the data is the slave computer in each case. The minicomputer will receive position coordinates, descriptions of symbols, and intensity from the host computer and reformat the data for the graphics displays. Hardware multiply and divide are required for the data formatting. The minicomputer hardware requirements are summarized as follows:

Word size	16 bits
Memory size, minimum	16K
Memory cycle time, maximum	1 microsecond
Input/output word transfer rate, minimum	400K words/second
Input/output word size	16 bits

6.4.2 Graphics Display and Keyboards

The display generator will have the capability of driving six CRT displays simultaneously with independent drawings. Also, the same display may be shown on any combination of CRT's. The display generator will have the capability of drawing points, vectors, characters, and circles. In addition, the display generator will have the capability of programmable intensity levels, and vertical and horizontal orientation of characters. Each of the vectors, circles, and characters will be programmable in size and position. The drawing rate or intensity compensation of the circles, vectors, and characters must be such that variations from the largest to the smallest will present the appearance of uniform brightness to the eye when each is at the same intensity setting. The display and keyboard hardware requirements are summarized as follows:

Display CRT, quantity	6
Display CRT size, minimum	12 in. diagonal
Phosphor type	P31
Spot diameter, maximum	0.020 inch
Display resolution, minimum	1024 x 1024
Vector drawing time full screen, maximum	70 microseconds
Vector drawing rate, minimum	0.07 inch per microsecond
Character drawing time average small character, maximum	10 microseconds
Character height, minimum range	0.15 in. to 0.50 in.
Programmable character sizes, minimum	4
Aspect ratio	1.2 - 1.5
Random beam positioning time, maximum	15 microseconds
Refresh rate, minimum	40 per second
Character set capability, minimum	64 symbols
End point closure between two vectors, maximum	0.05 inches
Intensity levels, minimum	4
Keyboard character, minimum	54
Function keys, minimum	10

DEVELOPMENT OF SIMULATION COMPUTER COMPLEX SPECIFICATION

6.4.3 Minicomputer to Host Computer Interface

An interface is required to transfer sixteen bits of parallel data both directions between the host computer and minicomputer. All control logic to effect the data transfer and to assemble and disassemble data words, as required, will be incorporated. The interface will transfer the most significant sixteen bits of any host computer data word and disregard the remaining bits. The data output is controlled from the transmitting computer. The interface will translate the request for data transfer as a request for block transfer. The amount of data in the block is determined by the transmitting computer. The computer receiving data is the slave computer in each case. The interface will handle all required communication between the host computer and minicomputer. In addition, the interface will receive the frame sync output from the real-time clock, described in Section 6.1.2, and interrupt the minicomputer upon receipt of this signal. Line drivers and line receivers are required at both the minicomputer and host computer. The data transfer rate will be a minimum of 200K words per second, and the maximum distance of the data transfer will be 100 feet. Other characteristics of the interface such as voltage levels and timing are a function of the minicomputer and host computer. The minicomputer manufacturer will supply the interface as part of the total package. Some development may be required; however, the development risk is negligible because many simulator systems have been implemented previously.

6.5 HOST COMPUTER TO MISSION CONTROL CENTER INTERFACE

An interface is required to convert parallel data to/from the host computer to serial data in the Mission Control Center telemetry format. All control logic to effect the parallel to serial and serial to parallel conversion and provide control signals to the host computer will be incorporated. The interface will be located adjacent to the host computer; therefore, line drivers and line receivers compatible with the MCC inputs and outputs are required. The input/output and control characteristics will be determined by the host computer selected. The output serial data rate is estimated to be 90K bits per second.

DEVELOPMENT OF SIMULATION COMPUTER COMPLEX SPECIFICATION

6.6 OPERATIONAL VERIFICATION

Automatic checkout will be used to verify the host computer to simulator crew station DCE including the minicomputer, minicomputer to analog/digital interface, digital-to-analog converters, and analog-to-digital converters. Automatic checkout is accomplished by closing the loop between the analog outputs and the analog inputs. The minicomputer will output data in steps to the digital-to-analog converters; read and verify the results through the analog-to-digital converters thereby closing the loop. The additional hardware required is a multiplexer and analog-to-digital converter for inputs and 68 additional digital-to-analog converters for outputs. This additional hardware will allow the loop closure for all analog inputs and outputs.

In the host computer to simulator crew station interface, automatic checkout provisions will not be required for the digital outputs and inputs at the data conversion equipment interface. If it is required, the proper place to close the digital loop is in the simulator. By far, most of the failures in the digital data chain will occur in the simulator; since most of the circuitry is located there. The failure rate for the digital transfer is very low in the interface, and considerable hardware and increased interface complexity is required for automatic checkout.

Diagnostic routines in all the minicomputers will verify each minicomputer subsystem and minicomputer peripheral units. Also, the routines will verify the real-time clock and, in conjunction with the host computer, will verify all the interfaces between the minicomputers and host computer. Also, the diagnostic routines will check the interfaces to the flight computers with the flight computers active. Also, the routines will check the interactive graphics in the instruction station by requesting displays through the keyboards and checking the displays.

6.7 GENERAL MAINTAINABILITY AND ENVIRONMENTAL REQUIREMENTS

The equipment will be capable of being repaired by replacement of easily removable circuit modules, cards, or components, wherever practical. Access will be provided to wire-wrap pins for maintenance and troubleshooting. Test points, adjustment points, cables, connectors, and labels will be accessible and visible during maintenance. Structural members or other mounted components of the unit or chassis will not interfere with access to components, connectors, or connections. Access doors, panels, or covers will open without interference or removal of items, wherever practical. Guards and shields will be provided, as necessary, to protect personnel and equipment.

Labels will be provided to identify and locate internal test points, adjustment points, connectors, and cables. All parts, subunits, and units likely to be replaced will be marked for identification. Markings will correspond with part numbers and reference designations used in assembly drawings and schematics. The equipment will be designed for a useful life of 10 years of cyclic operation when supported by maintenance. All parts, subassemblies, and assemblies will be interchangeable with other parts, subassemblies, and assemblies having the same part number. Any special test equipment, checkout boxes, special tools or extender circuit boards other than standard laboratory test equipment will be provided by the vendors.

The equipment is required to operate in the following environment:

Temperature, operating	10°C-50°C
Relative humidity without condensation	20% to 90%
Electrical power	208/120 VAC \pm 5% 3Ø/1Ø, 60 Hz \pm 0.5 Hz 480/277 VAC \pm 5% 3Ø/1Ø, 60 Hz \pm 0.5 Hz

The equipment will not be effected by radio frequency interference when installed in a normal laboratory environment without screen room protection. In addition, the equipment will not be permanently damaged by electrical power failures and electrical transients to \pm 25% of nominal voltage. The equipment will not be permanently damaged when subject to a non operative temperature of -10°C to +60°C.

DEVELOPMENT OF SIMULATION COMPUTER COMPLEX SPECIFICATION

6.8 ACCEPTANCE TESTING

Closed loop testing of the systems should be demonstrated using diagnostic programs in the minicomputers. All operating subsystems will include as applicable: addressable memory, arithmetic unit, operating registers, operating panel, real-time clock, paper tape punch, paper tape reader, magnetic tape units, and input/output channels. All diagnostic routines delivered with the system must be demonstrated. Diagnostic routines must also be required to test the hardware microprograms and the graphics displays.

The minicomputer to analog/digital interface must be tested in conjunction with the data conversion components. Closed loop testing using automatic checkout with the minicomputer must be used to test the analog outputs and analog inputs. The minicomputer must also be used to test the digital outputs, digital inputs, and addresses for each. Special checkout boxes or equipment will be provided by the vendor for monitoring digital outputs and providing digital inputs for the digital data checkout. In addition, cables, connectors, and other equipment as required must be provided by the vendor to implement the checkout. Diagnostic routines in the minicomputer will test each analog output and analog input in turn by outputting consecutive words in steps of one binary bit and verifying that each word is within its stated accuracy through the analog input. A different value for each data word will be output to each digital-to-analog converter during each output pass. This eliminates incorrectly addressed units replying with the correct input.

The minicomputer to host computer interfaces will be tested by connecting the interfaces to the host computer and transmitting data between the computers. The real-time clock interrupts can be tested by programming period and delay information and interrupting the minicomputers and host computer through the interfaces.

To test the graphics system, a block of data must be input through the keyboard, displayed and edited, and transmitted to the host computer. In response, the host computer will output displays to each of the CRT's. The real-time clock interrupt can be demonstrated by interrupting the minicomputer. The host computer must be operational and contain appropriate software.

The flight graphics interfaces will be tested by transmitting blocks of data from the host computer through the minicomputer to each flight CRT. The host computer and flight electronics must be operational and contain appropriate software.

6.9 RELIABILITY

The man-time-to-failure requirements for the various subsystems of the data conversion equipment have been defined and are presented in Reference 20. The input of configuration on the reliability of the various data paths will require further consideration at the time specific designs are established by appropriate vendors.

Section 7

FLIGHT COMPUTER EMULATION REQUIREMENTS

The use of actual flight software in the Shuttle Mission Simulator has been established as highly desirable, if not an absolute requirement. The Simulation Software Sizing Report, Reference 1, analyzed the feasibility of a bit-by-bit interpretive simulation of the flight computers by the host computer. The conclusion reached was that this was not a feasible approach for a realtime simulator because of the excessively high instruction rates required. The remaining alternatives are the use of the actual flight computers or a hardware emulation of them. The reference simulator configuration assumes the use of flight computers or their hardware equivalent. This section presents the requirements for microprogrammed computers to emulate the Shuttle flight computers presently being considered. The use of microprogrammable computers was considered a likely necessity for achieving minimal cost hardware. Data on candidate microprogrammable computers and an evaluation of their capability will be presented in the Background Survey, Reference 22.

Requirements for microprogrammed computers to emulate the flight computers were established by review of the general characteristics of microprogrammed computers currently available as well as review of Shuttle candidate flight computers. This information was then used to establish requirements for the microprogrammed emulator.

7.1 CHARACTERISTICS OF MICROPROGRAMMED COMPUTERS

Microprogrammed computers are different from computers with hardwired instruction sets with respect to their architecture. They offer specific advantages for certain applications, including emulation of other computers. They impose a requirement for developing the microcodes which give them their capability. These factors are discussed below.

7.1.1 Architecture

A microprogrammed machine differs from other types of computers in that its control signals are derived from a memory rather than from hardwired logic circuits. This memory stores the microinstructions which control the CPU and main memory. Often the I/O section is given its own memory and control.

F

The usual microprogrammed machine CPU has an assortment of registers and counters, an arithmetic and logic unit, and several data multiplexers to interconnect them. The microinstructions control the data multiplexers, arithmetic unit and registers to transfer data within the CPU or to and from the main memory. Each control signal in the machine is decoded from a few bits of the microinstruction register. A status register provides a set of bits which may be tested or selected for use by the microinstructions. These include an overflow bit, carry out bits, condition bits, sign bits, interrupt flag bits, and several (micro)programmable flag bits. Most microprogrammable machines allow a return address to be saved in a special register or to be pushed onto a stack for later use, so that microcode subroutines can be used without a time penalty. Machines with shorter microinstructions and faster cycles may require an extra step to save a return address.

7.1.2 Purpose and Advantages of Microprogramming

The flexible control of the interconnections and the programmed interpretation of instruction codes allow one hardware design to implement different instruction sets or to add special instructions at low cost. Machines with read-only control memories are fixed in their instruction set at the time of manufacture unless the read-only memory is replaced. A read-write control store allows the user to write microprograms and alter his machine.

The microprogrammed computer can be used to reduce cost by simplifying the logic design of a machine. It may be used to implement instructions such as searches and compares or block moves, which are complex if done in hardware. Some machines are built to emulate, i.e., perform like, an existing machine in order to make use of existing software. If the original machine is several years old, a new microprogrammed emulator may even be faster.

7.1.3 Emulators

An emulator must duplicate the behavior of every machine instruction of the original, preferably down to the last detail of floating point roundoff. The emulator's microprogram reads machine language instructions from memory into a register. It then uses bits of the instruction to form an address for the control memory, to which it transfers control. A routine is stored there which, by transferring data, emulates the instruction coded in the target machine's language. Each instruction of

DEVELOPMENT OF SIMULATION COMPUTER COMPLEX SPECIFICATION

the emulated machine is implemented as a microcode routine. The routine leaves the machine in a state corresponding to the state that executing the instruction leaves the original machine.

Each addressing mode would be done in a microcode subroutine. Instruction decoding can be done either by hardware or by microinstructions. Usually the operation code portion of the instruction register is able to index a microcode branch to the routine to process the operation code. Sometimes however, masking and shifting are required to form an address from an operation code. Almost any instruction set can be emulated if it is decoded by this technique, but it can be very slow. Some machines include a special decoding memory addressed by bits of the instruction and giving the micromemory address to execute.

Internally the input-output instructions are microprogrammed to operate as the original ones did, but the signal characteristics of the channels may be different to operate with other equipment.

7.1.4 Microprogramming

Writing the microinstructions for the machine is similar to machine language programming. The microprogrammer will require a more detailed understanding of the hardware and its timing plus detailed knowledge of the machine language of the machine to be emulated (target machine).

The microprogram is called the "firmware", as it is intermediate to software and hardware. Microcode assemblers which simplify microprogramming are usually available. A microinstruction is quite a bit more complicated than a machine language instruction. The words are longer (32 to 120 bits) and divided into many fields of one or more bits. Each field controls a particular part of the machine, such as a register selector, a counter, or the arithmetic unit. The meaning of some fields may be altered by the content of other fields, giving several formats for microinstructions.

A software simulator is usually used to debug microcode by tracing its execution. This is necessary if the front panel controls do not allow examination and stepping of the microprogram execution.

FINAL REPORT

DEVELOPMENT OF SIMULATION COMPUTER COMPLEX SPECIFICATION

MDC E0857
29 JUNE 1973

To achieve the maximum speed, each microinstruction should perform as many functions as possible. Thus one may be able, for example, to start a memory cycle, increment the program counter, test for an interrupt, and set a register with one microinstruction. The more microinstruction cycles per main memory cycle, the less parallelism is required, and the easier it is to program.

DEVELOPMENT OF SIMULATION COMPUTER COMPLEX SPECIFICATION

7.2 REQUIREMENTS FOR A SHUTTLE FLIGHT COMPUTER EMULATOR

Two computers being considered for the Space Shuttle are the IBM AP-101, and the Singer SKC-2000. The characteristics of these flight computers were analyzed to find hardware features which may require special treatment in an emulator.

The two candidate flight computers are very similar in computer power, although their instruction sets are quite different. To emulate them in real time requires a state of the art microprogrammed machine designed to emulate arbitrary instruction sets efficiently. The following sections will discuss requirements for different aspects of the microprogrammed emulator computer. It is possible for superior performance in one area to offset weakness of another, but failure to meet a requirement indicates a problem area which may disqualify a candidate.

Since the flight computer will be required to accommodate the flight programs with a large margin, an emulator which has 75% of the flight computer's capability is assumed to be potentially suitable for the simulator.

7.2.1 Memory Requirements

Either a 16 or a 32 bit word length memory may be considered since 16 bit instructions and halfword data is used in both flight computer candidates. The bandwidth (bit per second transfer rate) of the memory must be close to that of the flight computer.

A cycle time of one microsecond for a 32 bit memory or less than 500 nanoseconds for a 16 bit memory is required. Protect bits should be available for each 16 bits to emulate the AP-101, or for each word to emulate the SKC-2000. Without protect bits exact software performance may occur, but cannot be guaranteed. Parity bits are desirable. The size of the memory has to be expandable to that of the largest Space Shuttle flight computer, 65K of 32 bit words. A halfword addressable memory then requires 17 bits for an address. Sixteen bit machines will need a way to select the memory bank without delaying the memory cycle.

7.2.2 Microinstruction Capability

Microprogrammable machines may be very different in their microinstruction length, cycle time, and parallelism. These quantities are traded off in the design of the machine to such an extent that meaningful requirements cannot be set on physical parameters alone. Instead, a performance standard is required.

The emulator should be able to decode a machine language instruction within a single microinstruction cycle. Testing bits serially is too slow to be feasible. To decode the AP-101 addressing modes branching must be done based on groups of bits from the middle of the instruction. Therefore arbitrary groups of bits from the instruction register should be able to be selected to index the next microinstruction. Other fields of the instruction register have to be transferred for use as file addresses or shift counts with one microinstruction and in parallel with other operations. The ability to repeat execution of a microinstruction for shift, normalize, or multiply operations is desirable to avoid loops of two or more microinstructions.

The emulator should be able to decode an instruction rapidly. Bit by bit (or group of two bits) testing is not feasible. To decode the AP-101 addressing modes, groups of bits from the middle of the instruction must be tested. Therefore an arbitrary group of bits from the instruction should be able to be made into an address for the micromemory with one step. Other fields of the instruction register have to be easily transferred for use as file addresses or shift counts. The ability to repeat execution of a microinstruction for shift, normalize, or multiply operations is desirable to avoid loops of two or more instructions.

The emulator microprogram must be flexible enough to perform a complete emulation of all available instructions to avoid the need to change the flight computer programs. Multiple I/O channels have to be emulated by the machine. The microprogram has to control the channels to emulate AP-101 channels or SKC-2000 DMA's. The interrupt systems are not a difficult problem, but enough interrupt lines have to be available in the emulator. Sixteen are required for the AP-101 and the basic SKC-2000.

7.2.3 Register Requirements

Register files are used in all practical microprogrammed machines for temporary storage. In this application enough register storage should be available to implement at least one set of CPU registers of a flight computer. (Eight in the AP-101 and 15 in the SKC-2000.) Preferably all four sets should be able to be assigned to file registers. The SKC's 256 word scratchpad memory may have to be emulated with a large register file if the emulator's main memory cycle is slower than the flight computer's.

7.2.4 Arithmetic Unit Requirements

The arithmetic unit should have sufficient hardware to keep the instruction execution speed near that of the flight computers. Special shift registers and counters that parallel the arithmetic unit are of use. Enough data paths are required to avoid extra register transfers and to allow transfers to be done in parallel. Since the emulators have, by definition, the same instruction set as the flight computer and will execute the same code, a comparison based on an instruction mix can be used to evaluate the CPU capability. The emulator must achieve a speed close enough to the flight computers' to leave an adequate margin for some expansion of the planned software loading. The flight computers will have twice the capacity they initially require, but the emulator can risk operating closer to its limits. Therefore 75% of the flight computers processing capability should be the lower limit acceptable.

7.2.5 Other Requirements

The micromemory should be field alterable to allow improvements and corrections to be made, since we want the flight programs to run without alteration. Also, interfacing the I/O may be easier if microprograms can be changed.

Support programs such as a microcode assembler and a simulator should be available. A micromemory loader may be stored in a read-only portion of micromemory or hardware loading may be used.

7.2.6 Emulator Requirements Summary

Quantitative hardware requirements for a flight computer emulator are summarized in Table 7-1. Characteristics of several currently available microprogrammable computer may be found in Reference 22.

Table 7-1
Requirements for Emulator Computer

FEATURE	REQUIREMENT
Word Size	16 or 32 bits
Memory Speed	1 μ sec for 32 bits
Parity bits	1 per word or halfword
Protect bits	1 per word or halfword
Memory size	65K of 32 bit words
Interrupt lines	At least 16
Register file size	At least 32-32 bit registers
CPU Speed	2.7 μ sec/instruction, weighted average
Decode instruction speed	1 microinstruction cycle

The following operational features are required:

- Field alterable micromemory
- Microcode assembler

Section 8

SYSTEM SOFTWARE REQUIREMENTS - TASK 1.4

This Section summarizes the results of a Task performed to define the System Software Requirements for the Shuttle Mission Simulator. System software has been defined, for purposes of this study, to consist of operating system software, software processors, utility routines and library subroutines. The requirements in each of these areas have been established by careful study of three sources of functional requirements. These sources consisted of the following:

- Current vendor software capability
- General requirements for real time simulation facilities
- Requirements peculiar to the Shuttle Mission Simulator.

A detailed review of current vendor system software capabilities and features was carried out as part of the Background Survey task. This effort established a comprehensive base of features generally available with computers of the performance capability required for the Training Simulation Computer Complex. These basic requirements which are generally recognized and available from most vendors are summarized in Section 8.1.

The more specific requirements attributable to operation of a real time simulation facility, as established by review of experience at Johnson Space Center and at McDonnell Douglas Flight Simulation Laboratory, are summarized in Section 8.2.

Finally, system software requirements peculiar to implementation of the Shuttle Mission Simulator have been established by analysis of the expected software loading and review of the planned hardware interfaces. These SMS requirements are summarized in Section 8.3

A more detailed presentation of the System Software Requirements may be found in Reference 24.

8.1 GENERIC SYSTEM SOFTWARE REQUIREMENTS

The generic requirements have been established by careful review of the vendor documentation for system software available for the computers that are likely candidates for the Training Simulation Computer Complex. The specific operating systems reviewed were IBM's OS with RTM (Real Time Monitor); CDC's SCOPE 2.0 for the CYBER 70/MODEL 76; UNIVAC'S EXEC-8; and Texas Instruments system for the Advanced Scientific Computer. The language processors, utility programs and library routines associated with these operating systems were also reviewed.

8.1.1 Operating System

The support of the anticipated mode of operation at the simulation facility will require a comprehensive operating system. The operating system must provide for full multi-programmed operation to perform simultaneous real-time simulation, batch and time-sharing operation. It must provide a comprehensive job processing system, accounting system and operator control capability. It must provide maximum user assistance in file assignment and manipulation, error processing and I/O control.

8.1.1.1 Multi-programmed Operation

The system will be utilized to run a mix of operational simulations, checkouts of simulations, time-sharing interactive jobs, and batch programs in a multi-programmed environment. Thus, full multi-programming with software and hardware safeguards against any program's interference with the status or operation of any other program is required. The system is required to optimize operation by expediting the throughput of short jobs where this does not interfere with established priority schemes, and by maximizing the utilization of all resources whenever possible. The system must provide a comprehensive performance data acquisition package for storing information related to a job's progress through the system so that examination of such data by users or vendor analysts may disclose areas of throughput enhancement and potential bottlenecks. The systems must also provide the capability to maintain at least two simulation jobs and one interactive processing control program capable of servicing 15 terminals active simultaneously in addition to an unspecified number of batch jobs.

Multi-tasking

In addition to the described multi-programming capability, a multi-tasking capability is desired. The system should allow a number of independent programs to operate in unison to accomplish the required simulation task. These programs should be able to access common data areas, to enforce dependencies concerning their sequencing with respect to each other, to operate at individual sub-priorities and to be initiated based on clock or external interrupts. This capability would enhance the system in reducing the design of the simulation executive to the level of merely initiating a separate program to perform the processing of the various frame repetition rates.

Multi-processing

The study of host computer vendor data has indicated that some of the systems may require multiple CPU's to process the simulation software load. Also, some of the processors which are capable of handling the load imposed by a single simulator may require extra CPU's to run additional simulators.

The OS therefore must be extendable to a multi-processor configuration without affecting the application software.

8.1.1.2 Priority Algorithm

Multi-programmed systems require a priority algorithm to allow users to define the relative importance of the submitted jobs. Since the detailed day-to-day operational requirements of the simulation complex are impossible to predict at this time, the algorithm should be site-modifiable. The priority algorithm should allow variable time-slicing within fixed priority levels to insure the availability of the system to all users. The time-slice interval should be program controllable. Operator modification of job priorities should be allowed from interactive terminals attached to a job as well as from the operator control stations.

8.1.1.3 Executive Requests

Program accesses to the operating system monitor must be of two types: calls which result in a different job step gaining control of the CPU and calls where the current job step retains control of the CPU. The first type is required for

efficient computer utilization by batch jobs and the second is required for the simulation jobs where control of the CPU must be retained for specific amounts of time. The full complement of normal executive requests should be available as well as special functions dealing with communication with the real-time equipment.

8.1.1.4 Job Initiation

Control of at least two concurrently executing real-time simulation jobs in addition to time sharing operation of 15 independent terminals is required. The job initiation processor should determine the resource requirements of all new jobs entering to prevent initiation of jobs detrimental to operation of currently active jobs.

The operation of a simulator also requires the job initiator to check job dependencies to ascertain successful completion of prior jobs. This feature is necessary in creation of specific programs and load files and the initiation of appropriate simulations in an environment where many sequential loads may have the same or similar identification.

8.1.1.5 Job Step Scheduling

The operation of the simulation complex requires concurrent processing of various jobs by the host computer. A comprehensive job step scheduling function is therefore required. The scheduling function must serve the facilities requirements for site-modifiable scheduling algorithm, for dedication of the system to the real-time task and for enhancing the throughput of batch jobs. It should require minimal operator intervention, though the operator must have the option of changing any job's priority.

8.1.1.6 Resource Allocation

The system must enhance operation of the simulation complex by allocating only resources specifically required to jobs. As the resource needs of jobs are in a constant state of flux not only during subsequent submissions, but also during the course of a job, the resource allocation function should be as automatic as possible.

Memory and CPU time slice requirements should be handled on a job step basis without programmer action. Both physical device and logical file I/O allocation must

DEVELOPMENT OF SIMULATION COMPUTER COMPLEX SPECIFICATION

be provided with default assignment of devices for logical files.

8.1.1.7 Queue Control

The simulation complex will have both local and remote job submittal stations. The queue control system should maintain appropriate input and output queues so that normal output destinations are linked with appropriate job submittal station. The system must give equal consideration to all input queues when selecting jobs.

8.1.1.8 Program Loading

The programs used in the simulation facility comprise a mix of system and user library routines, compilers, files and input stream decks. The system must provide for mixed loading from all sources, for generation of absolutized load files of production programs and for relocatable loading.

The system must also provide overlay capabilities. These should be implemented so that the real-time simulation program can call for overlays, but continue processing while the overlay is being transferred into active memory.

Loading may be initiated via explicit or implicit reference or via control cards.

If fragmentation of core occurs due to termination or rollout of programs, the system may compact core whenever no interference with the simulation program can occur.

8.1.1.9 Event Monitoring

The system must be capable of recognizing and responding to various internal and external events. The simulation facility will require the system to respond to simulation hardware, to local and remote I/O stations, and to interactive terminals.

The system should also permit the specification of limits for execution time, main storage, and number of lines of output.

8.1.1.10 Termination Processing

The normal termination functions supplied by current systems is required by the simulation complex. The system must provide for explicit and implicit deallocation of files and other resources used during the course of the job. The system must

direct output files generated during the course of the job to the correct physical destinations.

All statistical and job history information generated during the job must be entered into the system's accounting file and appended to the user's printed output.

An abnormal termination processing capability should be provided to allow the user to continue his job in case of such termination. Optional core and file dumps are required as is the option of initiating recovery processing.

8.1.1.11 System Startup and Initialization

The normal system startup and initialization techniques should be provided. Permanent files must maintain their identity over all possible types of system power-down and startup procedures.

The system should permit specification of device availability and controlled system reconfiguration during startup.

All system starts should be recorded on the accounting file along with associated initialization information. The system should request time/date specification.

8.1.1.12 Job Control Communication

The system should provide for comprehensive job control from a number of sources. The operational mode of the simulation facility requires that the user be able to control his job from interactive terminals during both batch and real-time type operation.

8.1.1.13 I/O Control

The system is required to provide control of all hardware devices supplied as part of the main computer. The system should provide the maximum amount of default aids to the user to relieve him of unnecessary unit/device allocation. The system should perform all necessary blocking/deblocking, channel, device and buffering control and should provide all required data code translation. It should provide various modes of data organization control, including random and sequential processing. It should also provide full remote terminal support.

DEVELOPMENT OF SIMULATION COMPUTER COMPLEX SPECIFICATION**8.1.1.14 Resource Status Modification and Interrogation**

The system should provide all the necessary resource control aids for the operator to allow for minimal time loss in reconfiguring the system in event of hardware failures. The system should provide the operator with a list of alternatives resources to select from in replacing downed units. All allocations which can be resolved by the system according to installation parameters should be initiated and a message given to the operator for information purposes.

8.1.1.15 Recovery Processing

The system should provide a checkpointing method to allow users to periodically establish the status of long runs for restarting in case of equipment failure. The checkpointing should be initiated by interactive request, by program request or automatically based upon processing time. The entire status of the job should be retained to ensure error-free restarts.

8.1.1.16 Restarting

Provisions must be made for the restarting of checkpointed programs. The batch programs should be restarted automatically upon system reload, however the real-time simulation programs should be held until requested by operator or user command.

8.1.1.17 File Handling

The programs and data used in the simulation facility will reside on a set of user files. The system is required to provide comprehensive file handling services to aid the user in efficiently accessing his data.

8.1.1.18 Timing Services

The system must provide various timing services to aid users in checkout of simulation programs. Real-time clock and interval timing services should be provided.

The system should provide date and time conversion facilities.

The system should provide for task initiation, interruption and suspension on specified time basis.

8.1.1.19 Testing/Debugging Service

Since much of the operation of the simulation facility consists of developing and debugging complex real-time simulation programs, a comprehensive set of testing and debugging services is required. These should link to the software processors and should be available both in batch and real-time operation.

8.1.1.20 System Test Mode

The system should also provide an interactive test mode to allow on-line testing of all hardware. This mode should also allow for testing of simulation hardware either with programs supplied by the vendor or by allowing of addition of new test routines to the test program library.

8.1.1.21 Logging and Accounting

All information which can be gathered by the operating system concerning jobs and job steps is of extreme value to users of both batch and real-time operations. It can aid in pinpointing inefficient operation and trouble areas. If possible, the operating system should gather statistical information at both job and job step level and provide this information to the user. Preferably this should appear as a single report per job appended to the users printed output.

8.1.2 Software Processors

This section outlines the requirements to be met by various software processors. The requirements as given here are of a general nature and reflect the current state of host computer vendor-supplied software processing systems.

In general, the greater the number of software processors supplied by the host computer vendor, and the greater the number of processing options and functions each fulfills, the greater the utility of the system to the simulation complex.

8.1.2.1 Compilers and Assemblers

The software to be used in the simulator complex operation will to the greatest extent be developed and checked out on the simulator complex equipment. However,

DEVELOPMENT OF SIMULATION COMPUTER COMPLEX SPECIFICATION

a considerable amount of software will come from simulations currently under development as well as from the Software Development Laboratory for the Shuttle Flight Software.

Therefore, most of the software development will be done in higher level languages, and to allow for use of the programs on any of the systems, these systems must have compatible Software Processors. The required processors are FORTRAN, COBOL, and possibly PL1. The processors should adhere to the ANSI standards for FORTRAN and COBOL (References 25 and 26).

Though assembly language programs cannot be compatible, they too can meet certain standards. Some of these should be FORTRAN compatible COMMON and DATA structures and provision for subroutine linkage with higher-level processors. They should also contain coding aids such as a complete set of MACROS for all executive requests as well as complete MACRO capability.

Efficiencies

The modes of operation contemplated for the simulation complex require that the available software processors be supplied in two versions or selectable processing modes. This is desirable since two classes of operations exist.

Therefore, where possible, the software processors are required to have multiple optimization levels. They should be able to restructure source code to provide more efficient logical organization and/or hardware utilization. The high level of optimization should include most or all of the currently used optimization techniques. There should be as many intrinsic functions as possible to provide the high speed of in-line functions.

Data Handling

The large amount of Boolean data required for simulation operation requires a set of operators for easily manipulating bit and other data in the same expressions. For efficient object code these operations should be handled by in-line code if possible. The operations of AND, OR, Exclusive OR, NOT, SHIFT, and bit packing/unpacking into data words are required.

8.1.2.2 Editors

Text and linkage editors are required to allow operations to be performed upon source and binary data to build files of simulator programs for subsequent use.

8.1.2.3 Error Diagnostics

Each software processor must have a comprehensive, well documented set of error messages to reduce debugging effort. These diagnostics should be in an easily decipherable code, preferably some form of shortened and compacted English, to minimize literature search time. The types of inconsistencies checked for should

8.1.2.4 Debugging Features

In addition to the error diagnostics, the software processors should supply linking to system supplied trace and dump features to aid in program debugging. The system should allow utilization of these features by real-time programs by overriding diagnostics due to the violation of real-time operation rules.

8.1.3 Utility Programs

The system should provide a comprehensive set of utility routines to aid in the operation of the simulation facility. All standard utility programs available for the system should be supplied in addition to certain specific simulation required software. The following utility programs are required.

- File Copy and Positioning
- Library Operations
- System File Builders
- Maintenance Programs

8.1.4 Subroutine Libraries

The subroutine library, a part of the normal system library, should include the normal support software supplied with the system. A standard set of mathematical, I/O and software processor functions is required. The format of the library subroutines should be such that routines can be accessed freely from code generated by all available software processors. Naming conventions should leave all normal alphanumeric combinations free for user assignment, with the exception of such recognized standards as SIN, COS, SQRT, etc.

DEVELOPMENT OF SIMULATION COMPUTER COMPLEX SPECIFICATION

8.2 GENERAL SIMULATION SUPPORT REQUIREMENTS

The requirements stated in this section were gathered through review of operational experience at NASA and MDC simulation facilities. Reference 24 presents the results of a review of operations at the McDonnell Douglas Flight Simulation Laboratory, St. Louis, Missouri. Other sources included experience of MDAC-E personnel at the NASA-JSC Bldg. 35 simulation facility, and interviews with NASA personnel working with other simulators.

The requirements established from these sources deal with hybrid simulation control, OS response time, Real-Time I/O operations (hybrid I/O, overlays, data storage), special checkout systems, etc.

8.2.1 Operating System

The OS requirements in this section deal specifically with the special requirements associated with simulator operation. The monitor loop (or interrupt path) must provide sufficient system response to perform switching to the real-time task, execution of a high computational load, initiation of diverse I/O operations, and return to normal processing within a frame time of 40-50 ms. Therefore the system must perform at various levels of response for real-time, time-sharing demand processing, and batch operation to provide both the depth of services needed for various general computation tasks and the response required by real-time applications.

The interactive terminal functions are desirable and should be provided in some form. However, they stem from specially developed programs for the CDC 6000-series computers which can relieve the OS overhead problem by performing much of the required processing in PPU's. On systems where the entire processing load must be handled by the CPU(s), many of these functions which involve high repetition rates or computational loads should be used judiciously.

8.2.1.1 Monitor Response

In general, operation of real-time problems requires that the OS respond to the demands of the real-time equipment much faster than in a batch environment. Experience at various simulation facilities has shown that of major hardware manufacturers, both CDC and IBM have had to fine-tune their monitors to

provide the required response.

In the case of the CDC 6400 computer at Bldg. 35 (NASA-JSC), a second PPU in addition to the normal monitor PPU has been assigned the task of responding to real-time program interrupts and requests. IBM, on the other hand, has a special routine loaded for real-time interrupts for immediate processing to eliminate the excessive overhead associated with normal interrupt processing.

Since the operational load of the TSCC is such that it will operate at the limits of most currently available systems, some scheme of providing nearly instantaneous response is required. Without such enhancement, the system might not be able to process the simulation load either due to the imposed overhead, or due to the fact that delay in notification of a frame start would delay the processing to such an extent that the results would appear too late to influence external systems as required. As a rule, the overhead associated with all OS processing support to the real-time simulation should not exceed 10% of the total frame time. This requirement must be met in such a manner that all delays that may be encountered which would extend the actual frame time should be counted in the 10% time limit.

8.2.1.2 Priority Expediting

Operation of all types of batch and interactive programs in addition to real-time simulations requires that some mode of priority expediting be implemented in the OS. This should include automatic rollout of jobs to allow the high-priority real-time simulation programs to begin processing as soon as they appear on some input queue. In this way, training personnel's time will not be wasted by waiting for some low priority job to finish and release resources before they can begin a training simulation run.

8.2.1.3 Special Executive Requests

Certain additional executive request functions are required in simulation environments. These are concerned with simulator operation control and with timing information. An executive request for cumulative CPU time used by a job step is desirable. If possible, it should be initializable by a single request and the cumulative job step time should be placed into a specified memory location each time the requesting job is re-activated.

DEVELOPMENT OF SIMULATION COMPUTER COMPLEX SPECIFICATION

8.2.1.4 Job Initiation

The operation of a simulation complex requires that certain jobs, such as simulation and time sharing jobs with special expediting priorities, be started as soon as they appear in the system's input stream. Therefore the job initiation processor is required to recognize certain expediting priorities and to initiate necessary actions to free resources required by the jobs requiring expediting. These actions would include suspending and rolling out, and in cases even checkpointing, of lower priority jobs.

8.2.1.5 Job Step Scheduling

Simulator operation requires the job step scheduling function to be able to treat jobs on an individual basis depending on priority and other information. Since an impression of continuity for the trainees must be preserved, no delays due to roll-out or other interference with the operation of real-time jobs should occur. When delays due to unforeseen causes do occur, the system should inform the simulation job and enter any applicable data in the job's log. The simulation program should be able to set the threshold levels at which this condition is reported.

8.2.1.6 Resource Allocation

The real-time simulation problem must maintain control of the computer system for specific amounts of time, therefore it must be exempt from time-slicing except when other real-time simulations are concurrently active. Where more than one simulation is active the system should perform time-slicing only when job input parameters request it.

The system must also prevent roll-outs and storage moves involving real-time simulations to the benefit of lower priority jobs unless such operations do not interfere with the operation of the simulation job.

8.2.1.7 Simulation Operational Control

Special control software is required to initialize and operate the simulation equipment. In the previously mentioned Bldg. 35 simulation facility a PPU is assigned to handle the interface between the CDC 6400 and the simulation equipment. All functions during actual operation of the simulation are addressed to this PPU for processing. Special control cards initialize the operation of this PPU.

The system software is required to perform the functions of initialization, of analog and digital I/O, of analog scaling, and of interrupt recognition and real-time task start.

8.2.1.8 Real-Time Overlay

In many instances it is possible to partition the simulation problem into various segments or phases for either or both of two reasons:

1. There may be corresponding phases in the real system which provide natural points where a new overlay may be loaded in the simulation, and
2. the differential between human response time and computer speeds may be large enough to minimize the effect of overlay loading on the effectiveness of the training.

The basic difference in the mode of the overlay loading between real-time and batch processing is that the real-time program must retain control of the processor during overlay loading. This will result in the apparent continuity required by the training simulation while a new segment is loaded.

The system must allow the real-time program to request an overlay, to continue processing while the overlay is being loaded, and must notify the real-time program when overlay loading is complete.

8.2.2 Software Processors

The additional Software Processor requirements stem from the special operating requirements of simulation facilities.

The compilers should be able to function at multiple levels of optimization to satisfy requirements for fast compilations for checkout of initial program development and slower compilations but highly optimized machine code for the decks to be assembled into checked-out simulations. Also the compilers should provide special capabilities needed in a simulation environment such as the bit handling functions needed to process discrete data.

8.2.2.1 Bit Handling

A large portion of simulation problems involves handling of discrete data. This

DEVELOPMENT OF SIMULATION COMPUTER COMPLEX SPECIFICATION

data is normally received from and transmitted to the simulation hardware in packed form, but for efficient operation must be unpacked to allow testing and setting.

Reference 2 shows that a total of 8800 discrettes are transmitted per second (Module 1.S.4 - Hybrid Interface). For the reference computer it has been determined that 6 instructions are required per bit for either packing or unpacking discrettes, implying an excess of 50K instructions per second involved in this task.

The compiler should produce code that equals or betters this performance. A preferable optional solution would be an addressing capability which would allow testing, setting and resetting of packed discrettes, thereby eliminating the packing/unpacking operation.

8.2.2.2 Special Extensions

The simulation programs must be able to perform I/O operation in a buffered mode, continuing processing without waiting for completion of the I/O operation. Transfer of performance monitoring data to mass storage or tape for later data reduction is a common application of this function. Normally this function is performed by the BUFFER OUT/BUFFER IN statements which are a commonplace extension to the standard FORTRAN.

Other useful extensions that are required are ENCODE/DECODE, ENTRY, IMPLICIT, NAMELIST, and OVERLAY. As these functions are currently included in most of the FORTRAN processors currently available, it is only required that they be implemented according to common practice.

8.2.3 Utility Programs

A comprehensive set of Utility Programs should be included in the standard system software. In addition to the support defined in Section 8.1.3, there should be special maintenance support software for all interconnected units supplied with the TSCC. The maintenance programs should also provide for simple methods of adding additional checkout routines for future hardware.

Specific requirements have been established from the data provided in Appendix D, Reference 24, and inquiry into the Building 35 operation.

8.2.3.1 Simulator Hardware Maintenance, Test and Checkout

Reference 24 describes the checkout procedures used at McDonnell Automation to check the operation of various items of simulation hardware. Comparable programs to check out the simulation I/O interface are required. They should be accessible and controllable through the interactive terminals, should allow for use of either standard test values or values entered during the course of the test; and should provide for the recording of all data on an optional basis.

8.2.3.4 Interactive Control

The interactive control package DEANNA has been found to be a valuable aid in all phases of simulator operations on the CMPS and LMPS simulators in Bldg. 35 at NASA-JSC. The interactive control has been utilized to minimize simulation turn around while allowing the simulator operator maximum flexibility to obtain data and recover from abort conditions. The interactive control is a powerful checkout tool for the complex interruptable real-time simulation and has replaced two other user developed checkout programs.

The classes of capabilities provided by the DEANNA package are: Operator Aid, Job Control, File Scan, File Generation and Updating, File Utilities, and Terminate. The capabilities of primary use to the simulation user are Operator Aid and Job Control. The specific capabilities most commonly used are: accounting file monitoring, job monitoring, job control, data display and modify (octal and decimal), operating registers display, and programmed I/O. File utilities are also used to provide access to control card files for transfer to Job Control.

The DEANNA package provides additional capability to act as a remote computer console. This capability is valuable for systems checkout and verification in the normal multi-programming mode of operations. This capability is protected through the use of access codes. Specific capabilities include: control console commands, equipment monitoring and assignment, peripheral computer monitor, extended job control, and extended data modification.

DEVELOPMENT OF SIMULATION COMPUTER COMPLEX SPECIFICATION

8.2.3.3 Interactive Processing

Interactive processing at separate sites shall be provided. The processing shall be supported by the central complex on a demand basis and within the restrictions of providing real-time support. The interactive processing will be used to provide a direct means of software development and checkout.

Conversational Programming

The default access capability for the interactive processing shall be a conversational programming capability. A minimum set of operations shall consist of the following: automatic file definition for job and output files; semi-automatic sequencing for program input including format checking, a run mode; an alternate line step mode with display of intermediate results, and a program save command. The program language shall be FORTRAN compatible.

File Manipulation

File manipulation assumes the presence of a file or files established under the operations of Section 8.2.3.2. Two modes of operation are required: scan and update.

8.2.4 Library Routines

Appendix D of Reference 24 lists the special library routines developed at MCAUTO. These routines were designed either to add additional capability to the system or to provide improved performance over the standard system-provided routines. Recommended routines are described in Reference 24.

8.2.5 Software Management

The intended mode of operation and duration of existence of the TSCC is such that special requirements must be met by the software management method explicitly or implicitly supplied with the system software. The method must be extendable to simulation programs under development and in production use. Additional techniques familiar to the system vendor should be documented and supplied with the system. These should include the following:

- System Software Configuration Control
- Simulation Software Development Aids

8.3 SHUTTLE MISSION SIMULATOR LOAD DEPENDENT REQUIREMENTS

A digital computer system simulation has been used to supplement and verify engineering analysis of system requirements for the Training Simulation Computer Complex. Model development was facilitated by the use of a general computer simulation program, COMPSIM.

COMPSIM is a computer simulation tool which combines the general system modelling capabilities of discrete simulation languages such as GPSS and SIMSCRIPT with a set of functional operators specifically designed for the modelling of software-software and software-hardware interactions.

The COMPSIM models have been used to determine the number of interrupts, of I/O requests, and of operation of the scheduling algorithm for representative real-time and batch load mixes. These models cause continuously changing competition for CPU and I/O hardware resources. New active tasks are generated each time a clock interrupt is received or a batch job is processed by the batch monitor and resource allocator. Each active task operates in a unique manner as specified by its program model. When a task reaches highest priority, the scheduler gives it control of the CPU. After some processing time, the task will request I/O and, therefore, become inactive until the operation is completed. The next highest priority task will then get the CPU. When the I/O operation completes, the delayed task becomes an active task once more. If it has a higher priority, it will preempt the currently executing task.

The only portion of the anticipated SMS operation which has not been included in the simulation is the demand (time-sharing) processing. However, the software processor (FORTRAN compile) load due to time-sharing has been included in the batch model.

Table 8-1 summarizes the requirements established for system functions which must be executed at high rates.

Table 8-2 lists low-frequency occurrences such as compiler executions and job initiations.

Table 8-1

HIGH FREQUENCY EXECUTIONS OF SYSTEM SOFTWARE FUNCTIONS

FUNCTION		EXECUTIONS/SEC.		
		1-CPU	2-CPU	4-CPU
INTERRUPT RESPONSE				
REAL-TIME	CLOCK	25	50	100
	I/O (DEDICATED CHANNELS)	342	342	342
	I/O (MASS STORAGE)	18	18	18
TIME-SHARING	EXTERNAL	6	6	6
	I/O	6	6	6
BATCH	I/O	7	7	7
		404	429	479
I/O INITIATION				
REAL-TIME	DEDICATED CHANNELS	342	342	342
	MASS STORAGE	18	18	18
TIME-SHARING		12	12	12
BATCH		7	7	7
		379	379	379
TASK INITIATION				
REAL-TIME		25	50	100
TIME-SHARING		6	6	6
BATCH		25	50	100
		56	106	206
SPOOLING				
CARDS READ TO DISK		20	20	20
LINES PRINTED FROM DISK		80	80	80
		100	100	100

Table 8-2

LOW FREQUENCY EXECUTIONS OF SYSTEM SOFTWARE FUNCTIONS

FUNCTION	MEAN TIME BETWEEN OCCURRENCES (SECONDS)
PROGRAM LOAD	312
JOB INITIALIZATION	91
RESOURCE ALLOCATION	68
JOB TERMINATION	91
COMPILATION	360
SOURCE DATA MANIPULATION	720

8.4 REQUIREMENT MODIFICATIONS FOR TWO SIMULATORS

The Training Simulation Computer Complex is required to support two simulators. This report has addressed the requirements for one simulator. Factors requiring modification to accommodate a second simulator are discussed below.

The requirements presented in the Generic System Software and the General Simulation Support Software sections are unaffected by the additional load imposed by a second simulator. The multi-programming and multi-processing requirements stated in Section 8.1.1.1 cover the effect of operation of multiple jobs, and Section 8.2.1.6 specifies the need for time-sliced operation when multiple simulations are operating concurrently.

The overhead associated with operating system servicing of additional jobs must be determined. Table 8-1 details the overhead operations associated with a single simulator operating at a 40 ms timeframe. For additional simulations the number of clock interrupts, task initiations, and I/O initiations must be determined and added to the values in the table.

Similarly, the additional batch loading if any must be accounted for.

Section 9

SIMULATION MANAGEMENT PLAN - TASK 1.5

This Section summarizes the results of the Task performed to define a simulation management plan to expedite and monitor the procurement, development, implementation and acceptance of the Shuttle Mission Simulator Complex at the Lyndon B. Johnson Space Center. The primary objective of this plan was to determine if any additional hardware and system software requirements exist for the computer complex.

Section 9.1 presents a description of the procurement and development activities for the computer complex, data conversion equipment, flight computer hardware or flight computer emulator, simulator hardware and software, system integration, and system verification. The major tasks associated with these activities have been collected and integrated into a Simulation Management Plan Master Schedule. The milestones, consistent with the anticipated NASA plans, were presented for the time period of July 1973 through January 1978 in Reference 27.

A Simulator Control Process was presented in Reference 27. These control requirements delineated procedures, forms and computer facilities which are required for implementation of an effective Simulator Control Process. These requirements were a principal source of the additional hardware and software requirements summarized in Section 9.2.

9.1 TASK DESCRIPTIONS

The computer hardware and data conversion equipment requirements reported in the Computer Complex Hardware Requirements Final Report, Reference 20, and the system software requirements reported in the System Software Requirements Report, Reference 23, have been assumed to define the computer complex configuration. The configuration for the Shuttle Mission Simulator discussed in Section 3 is assumed.

The following subsections discuss the tasks for the procurement and development activities associated with development of this configuration. Section 9.1.1 discusses the Computer Complex, Section 9.1.2 the DCE, Section 9.1.3 the Flight Computer/Emulator, Section 9.1.5 the System Integration, and finally Section 9.1.6 the System Verification.

9.1.1 Computer Complex Procurement

The major tasks to be addressed during the Computer Complex procurement and development activity are concerned with the host computer hardware and system software and the actual flight computer hardware and software. Following the acquisition period for the computer complex, the tasks of this activity will concentrate on the operations and maintenance of the host computer.

Acquisition of the actual flight software, and procurement of the actual flight computer hardware should be managed under this activity. Assurance of the timely delivery of this government supplied equipment and the control and delivery of current flight software are the tasks to be performed.

The Contractor should develop the plan for the Computer Complex hardware/software acceptance and validation. This plan must include a definition of the test data, test procedures, and documentation for the provisional acceptance testing and formal acceptance testing periods. The Contractor should submit a report describing this plan six months after contract award, and actively participate in the evaluation and execution of the plan.

9.1.2 Data Conversion Equipment Procurement Task

There are three options for procurement of the data conversion equipment: as part of the computer complex, as part of the simulator equipment, or separately. From

DEVELOPMENT OF SIMULATION COMPUTER COMPLEX SPECIFICATION

the Background Survey Final Report, Reference 22, most of the vendors of high quality data conversion equipment are small companies that manufacture only data conversion equipment or manufacture data conversion equipment in conjunction with minicomputer systems. The selection of the computer complex or simulator vendor will be determined almost solely upon their computer or simulator equipment since the data conversion equipment is a very small part of that total system. The quality or cost of any data conversion equipment proposed with these computer or simulator systems would have a negligible impact on the final selection of the vendor. Therefore, by combining the data conversion equipment with either the computer complex or simulator procurement, the data conversion equipment accepted will be that proposed by the winning computer or simulator vendor regardless of the relative merits or cost of the DCE. By procuring the data conversion equipment separately, many more vendors will have the opportunity to bid on the system. Also, the procurement will be determined solely on the merits and cost of the DCE. Therefore, the highest quality DCE consistent with cost should result.

Furthermore, the data conversion equipment between the host computer and flight computers, and the graphics display system for the instructor console requires specialized systems. Specifically, microprogrammable minicomputers may be required for the flight to host computer DCE and graphics generating equipment is required for the instructor graphics displays. The number of vendors manufacturing the above equipment is limited and generally the same vendors do not manufacture both. Therefore, for the maximum response among bidders, the data conversion equipment should be divided further into three separate packages.

The disadvantages of increasing the number of procurement packages include: coordinating the acceptance testing of the several vendors, coordinating the hardware interface requirements between the vendors, and generally overseeing several vendors instead of two or three. In order to coordinate these activities, a Data Conversion Equipment Manager, and staff are proposed. The responsibilities of this group shall include the development and procurement activities for the hardware and software of all the data conversion equipment. This group shall be responsible for coordinating the interface between the Computer Complex and Simulator development and procurement.

DEVELOPMENT OF SIMULATION COMPUTER COMPLEX SPECIFICATION

The requirements are specified for the data conversion equipment (DCE) which support the data paths between the host computer and simulator hardware, flight computers, instructor operator station, flight hardware graphics, and the Mission Control Center. Cost analysis and integration trade studies are recommended to evaluate the best procurement approach for the DCE.

The Contractors should develop and deliver three months after contract award a plan identifying the integration, acceptance, and verification testing procedures for the DCE. Active participation in the execution of this plan will be required from the Contractors.

9.1.3 Flight Computer/Emulator Procurement Task

The use of actual flight software in the Shuttle Mission Simulator has been established as highly desirable, if not an absolute requirement. The Simulation Software Sizing Report, Reference 1, investigated the feasibility of implementing actual flight software with an interpretive simulation and a functional simulation of the flight software. The conclusion reached was that the actual flight software in a flight computer or emulator was the most desirable due to the excessively high instruction rate requirements of the other candidates. Section of the actual flight computer hardware reduces the development and procurement activity to a minimum.

Selection of a flight computer hardware emulator results in a series of procurement and development tasks that must be investigated. Section 7 presents the Flight Computer Emulator requirements that have been identified during the TSCC study. Since these requirements are based on an analysis of the characteristics of two flight computers presently being considered for the Space Shuttle, a review of the requirements is recommended following final selection of the Space Shuttle flight computer. In preparation for the acquisition of the flight computer emulator, tasks must be performed including the hardware and software specifications, RFP preparation, proposal preparation, and vendor evaluation, selection, and negotiation.

FINAL REPORT

DEVELOPMENT OF SIMULATION COMPUTER COMPLEX SPECIFICATION

MDC E0857
29 JUNE 1973

The major tasks that must be performed during the acquisition of the flight computer emulators shall include but not be limited to:

1. Development (fabrication) of the emulator hardware
2. Development of the microprogram code
3. Development (fabrication) of the emulator DCE
4. Delivery and installation of the emulator hardware and microprogram code
5. Acceptance testing of the emulator hardware and microprogram code

If the front panel controls of the emulator hardware do not provide the capability to examine or single step the microprogram execution, a software simulator should be required to allow for debug execution of the microcode. No development time need be allotted for this software since it should be a standard FORTRAN software program available from the vendor.

Specific details describing the work breakdown structure and schedules for the subtasks similar to those presented above should be developed by the Flight Computer Emulator vendors and submitted for evaluation as part of their procurement packages.

The Contractor must be responsible for the preparation and delivery of a documentation plan to support the development and procurement activities. This plan must identify and describe all documentation items and their delivery schedule.

One month after contract award, the Contractor should submit a plan which identifies the delivery, integration, acceptance testing, and verification testing procedures. Active participation in the execution of this plan will be required from the Contractor.

The Simulator Contractor shall be responsible for the development (fabrication), delivery, installation, acceptance testing, and maintenance of the Shuttle Mission Simulator hardware and software. The hardware tasks include the design and

fabrication, the delivery and installation, and the provisional acceptance testing of the five major hardware items which include:

1. Simulator hardware (crew station)
2. Visual system
3. Instructor Operator Station
4. Motion Base
5. Manipulator System

The Simulator Contractor will elect to develop these hardware items in-house or possibly select a subcontractor(s) to do certain items. Whatever approach is selected, the Simulator Contractor shall be responsible for the timely delivery and installation of all the hardware items. System integration and verification support shall guarantee the interface between the different hardware items and the Simulation Software. A time phased delivery cycle is required for all hardware, thus providing a basic simulator capability early in the development cycle and progressing to the full capability simulator. The Simulator Contractor should be responsible for the coordination and integration of the DCE following provisional acceptance of that equipment.

During the Simulator Contractor selection, bidders should be required to define work breakdown structure, schedules, and task descriptions for accomplishing the Simulator hardware and software development.

As defined in the other development and procurement activities, the successful Simulator Contractor should develop and deliver to SMS Project Management for approval a documentation plan and a plan for the acceptance testing and verification of the Simulation hardware and software.

9.1.5 System Integration Task

Primary system integration activities would consist of system definition, system integration test requirements definition, configuration management and related management information system development and operation. The system integration group can supply the technical manager with total system configuration data, facility requirements, integrated and coordinated development of test schedules and can support the entire project with respect to documentation, configuration, and management information.

DEVELOPMENT OF SIMULATION COMPUTER COMPLEX SPECIFICATION

The system definition task initially would require assembly of all available data on SMS requirements and Shuttle Program training requirements. The system integration group can then generate the total simulator configuration trade study data required to enable the Project Manager to select the most desirable configuration and the associated distribution of hardware and functions between the major subsystems. For example, the definition of the interface between the simulator hardware and the DCE will require delineation. The responsibility for flight computer procurement and integration requires assignment. The amount of redundancy to be provided by flight hardware requires evaluation. After the Project Manager decides these issues and obtains the required approvals, the subsystem managers (i.e., computer complex, simulator, DCE) can proceed with their procurements. The system integration group will then focus on coordinating schedule, facility, data, and testing requirements.

The systems integration group can also be responsible for planning and supporting the tests required to assure successful integration of all subsystems. This will require definition of test requirements and then definition of temporary interfaces, scaffolds, for test purposes.

The systems integration group, because of its own information requirements, is the logical group to establish documentation requirements and change control procedures, as well as an adequate configuration management system. These procedures, requirements and systems should be established to support the simulator throughout its entire life cycle.

The system integration group is a strictly technical group, however, the implementation of a configuration management system will facilitate development and operation of a parallel management information system. Cost or staffing data and requirements should be established and provided by the Project Manager's staff which will be the prime user of this data.

9.1.6 System Verification Task

The Verification Task can include verification test planning and support, and all actual test operations activities. The primary verification tasks, early in the development, would be to support the integration staff with the execution of the provisional acceptance testing and to develop hardware and software testing tools and plans. Following the provisional acceptance testing, the verification group shall concentrate on the performance of the total system and integrated major subsystems testing. Identification of test operations, test data processing, and test documentation can be performed by the Verification group. Final acceptance testing of the total system capability to satisfy the specifications should be demonstrated by the Verification group through the execution of a mission simulation. The support of all subcontractors is required during the test.

DEVELOPMENT OF SIMULATION COMPUTER COMPLEX SPECIFICATION

9.2 COMPUTER COMPLEX HARDWARE/SOFTWARE REQUIREMENTS

Requirements have been identified in this Task which will be incorporated into the Computer Complex hardware and system software specifications for the TSCC. These requirements are summarized in Table 9-1. Presented in this table is a summary of the requirements versus the task or function of the Simulation Management Plan that they support.

The Simulator Control Process, Reference 27, established requirements for the following: A schedule processor is required to maintain a waterfall schedule and to manage the work breakdown structure. Bar charts can be prepared on the line printer or if desirable a plotter shall be required to prepare the charts. A COBOL language processor is required to support this function.

Management Information Systems of the type discussed in Reference 27, are available from almost all commercial computer manufacturers. The basic functions which are required include: 1) exception report preparation, 2) PERT/time software, 3) Task assignment management, and 4) cost accounting. A COBOL language process is required to support these functions. A standard line printer is required for report preparation of PERT chart information. Sort program software and other application programs are required to support the Management Information System. These support software programs minimize manhours required in the control and distribution of action items and manpower within the project.

Hardware and software requirements have been identified to support the Software Configuration Control function. An ANSI FORTRAN compiler and support software program which test for standards violation are required. Flowchart software and hard copy plotters are required to support the documentation process of the Simulation Software. Line printers may be used for the preparation of flowcharts. Although MDAC success with commercial flowchart software packages has not been encouraging, a flowchart program is useful and recommended. Text editing, file processing, and library update capabilities are required to be provided by the computer complex system software. Interactive display terminals are required to support the software development.

Table 9-1

Computer Complex Hardware/Software Requirements Summary

TASK/FUNCTION	COMPUTER HARDWARE REQUIREMENTS	SYSTEM SOFTWARE REQUIREMENTS	SPECIAL SUPPORT HARDWARE REQUIREMENTS	SPECIAL SUPPORT SOFTWARE REQUIREMENTS
Work Breakdown Structure Waterfall Schedule	Printer/Plotter	COBOL		Schedule Processor
Management Information System (MIS)				
Exception Report	Printer	COBOL		Sort Program
PERT/Time	Printer/Plotter	PERT Software		Applications Program
Task Assignments	Printer	COBOL		Sort Program
Cost Accounting	Printer	COBOL		Applications Program
Software Configuration Control				
Standards	Printer	ANSI FORTRAN Compiler		Testing Software
Software Development	Plotter Printer Interactive Terminals	Text Editing File Processing Library Update		Flowchart Software
Data Base Management	Plotter/CRT Printer	File Protection File Labeling Text Editing File Processing		Sort Programs Applications Program
Hardware Configuration Control				
Preventive Maintenance			Test Hardware	
Spares Control		Bill of Materials Processor		

Table 9-1

Computer Complex Hardware/Software Requirements Summary (Continued)

TASK/FUNCTION	COMPUTER HARDWARE REQUIREMENTS	SYSTEM SOFTWARE REQUIREMENTS	SPECIAL SUPPORT HARDWARE REQUIREMENTS	SPECIAL SUPPORT SOFTWARE REQUIREMENTS
Computer Operations Control				
Component Utilization Monitoring	Printer/Plotter	Accounting Software	Performance Measurement Equipment	Performance Measurement Program
Tape Library Maintenance	Printer	Utility Software		
Acceptance				
Computer Complex Vendor Evaluation/Selection				Operations Benchmark
Acceptance Testing			Test Hardware	Operations Benchmark Applications Programs Test Software
Verification	Hardware Self Test	System Manuals		Performance Verification Test Programs
Documentation	Printer Plotter Schematics Wiring Diagrams Maintenance Manuals	System Manuals Text Editing		Flowchart Programs Applications Programs Report Generation Software

Section 10

CONCLUSIONS AND RECOMMENDATIONS

The Training Simulation Computer Complex Study is now complete although the Shuttle Vehicle Program is still in its early development phase and subcontractors for some of the major subsystems remain to be announced. Throughout the study, a major objective has been to establish and maintain maximum visibility between the desired computer complex requirements and related Shuttle vehicle configuration requirements. The end items and supporting documentation delivered to the contract Technical Manager provide maximum visibility between software loadings desired and such factors as assumed simulator fidelity requirements, subsystem redundancy levels, Shuttle data processing system computation frequencies, and Shuttle Mission Simulator Training requirements. Consequently, the conclusions and recommendations summarized below emphasize those aspects of computer complex requirements that are particularly sensitive to the evolution of the Shuttle configuration and its associated training requirements.

10.1 CONCLUSIONS

The computer hardware, system software and data conversion equipment requirements have been established to support a comprehensive Shuttle Mission Simulator configuration. The modularity and detail of the data supplied facilitates estimation of requirements for modified or multiple simulator configurations with minimum effort. A computer program, PSYZR, has been developed and supplied to the contract Technical Manager to expedite estimation of software loadings for alternate configurations.

A FORTRAN operations mix, typical of JSC simulation software, has been derived for specification of CPU processing requirements. CPU requirements have been derived for single, dual, and quadruple processor configurations in terms of millions of FORTRAN operations per second, MOPS, as well as millions of instructions per second; MIPS. The requirements specified for the configurations studied are summarized below for each CPU and each memory:

FINAL REPORT

DEVELOPMENT OF SIMULATION COMPUTER COMPLEX SPECIFICATION

MDC E0857
29 JUNE 1973

<u>Configuration</u>	<u>Memory Words</u>	<u>MOPS</u>	<u>MIPS</u>
Single CPU/Single Computer	284000	2.10	8.4
Dual CPU With Shared Memory	284000	1.25	5.0
Four CPU With Shared Memory	284000	0.78	3.1
Dual Computer	213000	1.25	5.0
Four Computer	184000	0.78	3.1

These requirements are limited to processing of the simulator real-time and batch loads. Operating system overhead and memory requirements must be in added to these requirements.

The flight software loading has also been estimated and requirements for interpretive simulation, functional simulation, and emulation have been derived. Based on past experience and gross cost factors, use of actual flight computers seems most advantageous at this time.

The general requirements for system software as well as the requirements for operating overhead processing have been derived. The digital computer system simulation has established the requirements for high frequency system overhead functions as follows:

- 400 responses per second to interrupts
- 380 I/O initiations per second
- 56 task initiations per second
- 100 cards or lines of print per second spooling rate.

Execution of these operations is required over and above the simulation software and batch/interactive loads.

Data conversion equipment requirements have been derived for data paths between the host computer and the simulator crew station, between the host computer and the flight computers, between the host computer and the flight CRT electronics, and between the host computer and the Mission Control Center. Also derived were the requirements for the graphics display system of the instructor station, and for the real-time clock for simulation timing. Conventional and special purpose micro-programmed minicomputers are required for the data conversion equipment. Conventional minicomputers provide for data formatting and data distribution to the simulator crew stations, simulator graphics displays, and the instructor display

FINAL REPORT

DEVELOPMENT OF SIMULATION COMPUTER COMPLEX SPECIFICATION

MDC E0857
29 JUNE 1973

terminals. Special purpose, microprogrammed minicomputers are required for the necessary data formatting, storage, and interrupt processing for data transfer between the host computer and flight computers.

Simulation Management Plan requirements have been identified which support the procurement and development activities of the Shuttle Mission Simulator. Support software and applications software programs have been identified to support the Simulator development control process.

10.2 RECOMMENDATIONS

The following recommendations seem justifiable on the basis of the results of the Training Simulation Computer Complex Study:

- It is recommended that for the computer procurement, CPU speed requirements be specified in terms of an operations executed rate, MOPS, and that a benchmark program, such as included with the Training Simulation Computer Specification, be used as part of the performance demonstration.
- It is recommended that the use of flight computers for the SMS be reviewed shortly after selection of the flight computer and flight software contractors and again after selection of the simulator computer contractor.
- It is recommended that the software loading be reviewed for the effects of changes in subsystem redundancies, and flight computer computation rates when the Shuttle avionics have advanced further in the development cycle.
- It is recommended that the training requirements for the payload manipulation activities on the Shuttle Mission Simulator be reviewed and defined more explicitly. Altering SMS training requirements in this area could result in a significant reduction to the manipulator imposed 2.6 MIPS compute load during the orbital phase.
- It is recommended that the data conversion equipment be procured through separate, independent, competitive procurements.

B

Section 11

REFERENCES

1. Development of Simulation Computer Complex Specification - Simulation Software Sizing Report. McDonnell Douglas Astronautics Company - Eastern Division, MDC E0739, 15 January 1973.
2. Shuttle Mission Simulator Software Module Sizing Summaries. McDonnell Douglas Astronautics Company-East, MDC E0748, 15 January 1973.
3. Technical Proposal for Space Shuttle Program. North American Rockwell Corporation, SD 72-SH-50-3, 12 May 1972.
4. M. E. Fowler. Numerical Methods for the Synthesis of Linear Control Systems. Automatica, 1963, Vol. 1, pp. 207-225.
5. M. E. Fowler. A New Numerical Method for Simulation. Simulation, May 1965, Vol. 4, pp. 324-330.
6. M. E. Fowler and T. H. Witzel. All Digital Reentry Simulation. IBM Report No. 63-508-2, 27 March 1963, FSD Space Guidance Center, Owego, N. Y.
7. M. E. Fowler. Supersonic Transport Simulation. McDonnell Douglas Astronautics Company Houston Operations Simulation Technology Department Design Note No. 1, December 1969.
8. J. M. Hurt. New Difference Equation Technique for Solving Nonlinear Differential Equations. AFIPS Conference Proceedings, Vol. 25, 1964 Spring Joint Computer Conference, pp. 169-179.
9. An Introduction to Real-Time Digital Flight Simulation. IBM Manual E20-0034.
10. Numerical Techniques for Real-Time Digital Flight Simulation. IBM Manual E20-0029.
11. Application of Numerical Techniques to Flight Simulation. IBM Manual E20-8186.
12. J. S. Rosko. Digital Simulation of Physical Systems. Addison-Wesley Publishing Co., Reading, Massachusetts, 1972, pp. 318-351.
13. A. P. Sage and R. R. Burt. Optimum Design and Error Analysis of Digital Integrators for Discrete-System Simulation. AFIPS Conference Proceedings, 27, Pt. 1, 1965, pp. 903-914.
14. J. M. Smith. Software Design Using Sampled-Data Techniques. McDonnell Douglas Astronautics Company Houston Operations, Houston, Texas.
15. Documentation for MOPAS Subroutine RIGID. Chrysler Corporation Space Division, Technical Report TN-AP-69-408, August 1969.

FINAL REPORT

DEVELOPMENT OF SIMULATION COMPUTER COMPLEX SPECIFICATION

MDC E0857
29 JUNE 1973

16. Summary of Equations for Preliminary Shuttle Ascent Simulation - SACS Working Paper No. 2. McDonnell Douglas Astronautics Company, Memo No. E933-327, 25 January 1972.
17. Request for Computer Services - Six Degree of Freedom Elastic Body Wind Response Program. Chrysler Corporation Space Division, Memo, 2 April 1966.
18. Final Report. Space Transportation System Software Concepts Development Study. TRW Systems, 21455-6002-R0-00, 31 July 1972.
19. Contract NAS 0-14000, Orbiter Avionics System Baseline Revision. NASA Letter EJ-72-191, ME-12-198, 27 November 1972.
20. Development of Simulation Computer Complex Specification - Computer Complex Hardware Requirements Final Report. McDonnell Douglas Astronautics Company - East, MDC E0798, 30 April 1973.
21. R. T. Adams, W. H. G. Caldwell, Jr., and D. E. Carlton. Evaluation of Time-Sharing Systems - Benchmark. School of Information Sciences, Georgia Institute of Technology. Report GITIS-69-22, 1969.
22. Development of Simulation Computer Complex Specification - Background Survey Final Report. McDonnell Douglas Astronautics Company - East, MDC E0813, 29 June 1973.
23. Development of Simulation Computer Complex Specification - System Software Requirements Final Report. McDonnell Douglas Astronautics Company - East, MDC E0799, 30 April 1973.
24. Development of Simulation Computer Complex Specification - System Software Requirements Final Report. McDonnell Douglas Astronautics Company - East, MDC E0799, 30 April 1973.
25. ANSI Standard FORTRAN, X3.9-1966. American National Standards Institute, 10E. 40th St., New York, N. Y. 10016
26. ANSI COBOL, X3.23-1968. American National Standards Institute, 10E. 40th St., New York, N. Y. 10016
27. Development of Simulation Computer Complex Specification - Simulation Management Plan Report, McDonnell Douglas Astronautics Company - East, MDC E0824, 31 May 1973.
28. Don Smith, Jr., "An Organization for Successful Project Management," Computer Sciences Corporation, El Segundo, California, Spring Joint Computer Complex Proceedings, 1972.